(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
8 March 2001 (08.03.2001)

PCT

(10) International Publication Number
WO 01/16862 A2

(51) International Patent Classification[7]: G06F 19/00

(21) International Application Number: PCT/US00/40805

(22) International Filing Date:
1 September 2000 (01.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/151,818 1 September 1999 (01.09.1999) US

(71) Applicant: **CALIFORNIA INSTITUTE OF TECHNOLOGY** [US/US]; 1200 East California Boulevard, MC 201-85, Pasadena, CA 91125 (US).

(72) Inventors: **GORDON, David, B.**; 175 Beacon Street, Somerville, MA 02143 (US). **MAYO, Stephen, L.**; 530 S. Greenwood Avenue, Pasadena, CA 91107 (US).

(74) Agents: **TRECARTIN, Richard, F.** et al.; Flehr Hohbach Test Albritton & Herbert LLP, 4 Embarcadero Center, Suite 3400, San Francisco, CA 94111-4187 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
Without international search report and to be republished upon receipt of that report.

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 01/16862 A2

(54) Title: METHODS AND COMPOSITIONS UTILIZING A BRANCH AND TERMINATE ALGORITHM FOR PROTEIN DESIGN

(57) Abstract: The present invention relates to apparatus and methods for quantitative protein design and optimization. In particular, the invention describes the use of the Branch and Terminate algorithm in protein design.

METHODS AND COMPOSITIONS UTILIZING A BRANCH AND TERMINATE ALGORITHM FOR PROTEIN DESIGN

This application is a continuing application of U.S.S.N. 60/151,818, filed September 1, 1999.

FIELD OF THE INVENTION

The present invention relates to an apparatus and method for quantitative protein design and optimization. In particular, the invention describes the use of the Branch and Terminate algorithm in protein design.

BACKGROUND OF THE INVENTION

De novo protein design has received considerable attention recently, and significant advances have been made toward the goal of producing stable, well-folded proteins with novel sequences. Efforts to design proteins rely on knowledge of the physical properties that determine protein structure, such as the patterns of hydrophobic and hydrophilic residues in the sequence, salt bridges and hydrogen bonds, and secondary structural preferences of amino acids. Various approaches to apply these principles have been attempted. For example, the construction of α-helical and β-sheet proteins with native-like sequences was attempted by individually selecting the residue required at every position in the target fold (Hecht, *et al.*, Science **249**:884-891 (1990); Quinn, *et al.*, Proc. Natl. Acad. Sci USA **91**:8747-8751 (1994)). Alternatively, a minimalist approach was used to design helical proteins, where the simplest possible sequence believed to be consistent with the folded structure was generated (Regan, *et al.*, Science **241**:976-978 (1988); DeGrado, *et al.*, Science **243**:622-628 (1989); Handel, *et al.*, Science **261**:879-885 (1993)), with varying degrees of success. An experimental method that relies on the hydrophobic and polar (HP) pattern of a sequence was developed where a library of sequences with the correct pattern for a four helix bundle was generated by random mutagenesis (Kamtekar, *et al.*, Science **262**:1680-1685 (1993)). Among non de novo approaches, domains of naturally occurring proteins have been

modified or coupled together to achieve a desired tertiary organization (Pessi, *et al.*, Nature **362**:367-369 (1993); Pomerantz, *et al.*, Science **267**:93-96 (1995)).

Though the correct secondary structure and overall tertiary organization seem to have been attained by several of the above techniques, many designed proteins appear to lack the structural specificity of native proteins. The complementary geometric arrangement of amino acids in the folded protein is the root of this specificity and is encoded in the sequence.

Several groups have applied and experimentally tested systematic, quantitative methods to protein design with the goal of developing general design algorithms (Hellinga, *et al.*, J. Mol. Biol. **222**: 763-785 (1991); Hurley, *et al.*, J. Mol. Biol. **224**:1143-1154 (1992); Desjarlais, *et al.*, Protein Science **4**:2006-2018 (1995); Harbury, *et al.*, Proc. Natl. Acad. Sci. USA **92**:8408-8412 (1995); Klemba, *et al.*, Nat. Struc. Biol. **2**:368-373 (1995); Nautiyal, *et al.*, Biochemistry **34**:11645-11651 (1995); Betzo, *et al.*, Biochemistry **35**:6955-6962 (1996); Dahiyat, *et al.*, Protein Science **5**:895-903 (1996); Jones, Protein Science **3**:567-574 (1994); Konoi, *et al.*, Proteins: Structure, Function and Genetics **19**:244-255 (1994)). These algorithms consider the spatial positioning and steric complementarity of side chains by explicitly modeling the atoms of sequences under consideration. To date, such techniques have typically focused on designing the cores of proteins and have scored sequences with van der Waals and sometimes hydrophobic solvation potentials.

Recent studies using coiled coils have demonstrated that core side-chain packing can be combined with explicit backbone flexibility (Harbury et al., PNAS USA 92:8408-8412 (1995); Offer & Sessions, J. Mol. Biol. 249:967-987 (1995). In these cases, the goal was to search for backbone coordinates that satisfied a fixed amino acid sequence.

In addition, the qualitative nature of many design approaches has hampered the development of improved, second generation, proteins because there are no objective methods for learning from past design successes and failures.

Thus, it is an object of the invention to provide computational protein design and optimization via an objective, quantitative design technique implemented in connection with a general purpose computer.

SUMMARY OF THE INVENTION

In accordance with the objects outlined above, the present invention provides methods executed by a computer under the control of a program, the computer including a memory for storing the program. The methods comprise the steps of receiving a protein backbone structure with variable

residue positions, establishing a group of potential rotamers for each of the variable residue positions, wherein at least one variable residue position has rotamers from at least two different amino acid side chains, and analyzing the interaction of each of the rotamers with all or part of the remainder of the protein backbone structure to generate a set of optimized protein sequences. The methods further comprise classifying each variable residue position as either a core, surface or boundary residue. The analyzing step may include a Branch and Terminate (B&T) computation either alone or in combination with a Dead-End Elimination (DEE) computation. Generally, the analyzing step includes the use of at least one scoring function selected from the group consisting of a Van der Waals potential scoring function, a hydrogen bond potential scoring function, an atomic solvation scoring function, a secondary structure propensity scoring function and an electrostatic scoring function. The methods further comprise altering the protein backbone prior to the analysis, comprising altering at least one supersecondary structure parameter value. The methods may further comprise generating a rank ordered list of additional optimal sequences from the globally optimal protein sequence. Some or all of the protein sequences from the ordered list may be tested to produce potential energy test results.

In an additional aspect, the invention provides nucleic acid sequences encoding a protein sequence generated by the present methods, and expression vectors and host cells containing the nucleic acids.

In a further aspect, the invention provides a computer readable memory to direct a computer to function in a specified manner, comprising a side chain module to correlate a group of potential rotamers for residue positions of a protein backbone model, and a ranking module to analyze the interaction of each of said rotamers with all or part of the remainder of said protein to generate a set of optimized protein sequences. The memory may further comprise an assessment module to assess the correspondence between potential energy test results and theoretical potential energy data.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a general purpose computer configured in accordance with an embodiment of the invention.

Figure 2 illustrates processing steps associated with an embodiment of the invention.

Figure 3 illustrates processing steps associated with a ranking module used in accordance with an embodiment of the invention. After any DEE step, any one of the previous DEE steps may be

repeated. In addition, any one of the DEE steps may be eliminated; for example, original singles DEE (step 74) need not be run.

Figure 4 is a schematic representation of the minimum and maximum quantities (defined in Eq. 24 to 27) that are used to construct speed enhancements. The minima and maxima are utilized directly to find the $(i_u j_v)_{mb}$ pair and for the comparison of extrema. The differences between the quantities, denoted with arrows, are used to construct the q_{rs} and q_{uv} metrics.

Figures 5A, 5B, 5C and 5D depict several super-secondary structure parameters for α/β proteins. The definitions are similar to those previously developed for α/β proteins (Janin & Chothia, J Mol Biol 143:95–128 (1980); Cohen et al., J Mol Biol 156:821–862 (1982)). The helix center is defined as the average C_α position of the residues in the helix. The helix axis is defined as the principal moment of the C_α atoms of the residues in the helix. (Chothia et al., Proc Natl Acad Sci USA 78:4146–4150 (1981); J Mol Biol 145:215–250 (1981). The strand axis is defined as the average of the least-squares lines fit through the midpoints of sequential C_α positions of two central β-strands. The sheet plane is defined as the least-squares plane fit through the C_α positions of the residues of the sheet. The sheet axis is defined as the vector perpendicular to the sheet plane that passes through the helix center. Ω is the angle between the strand axis and the helix axis after projection onto the sheet plane; θ is the angle between the helix axis and the sheet plane; h is the distance between the helix center and the sheet plane; σ is the rotation angle about the helix axis. The super-secondary structure parameter values for native Gβ1 are Ω = -26.49°, θ = 3.20°, h = 10.04 Å and σ = 0°.

Figures 6A, 6B, 6C and 6D depict four supersecondary structure parameters for β/β protein interactions. Figures 6A and 6B are relevant to β barrel proteins; Figure 6C is relevant to β-sheet interactions. Figure 6A shows only three strands, and depicts R, the barrel radius; α, the tilt of the strands relative to the barrel axis; a, the distance from C^α to C^α along the strands; and b, the interstrand distance. Figure 6B shows the twist and coiling angles of the β-sheet, with residues A, B and C from one strand, residues D, E and F in strand 2, and residues G, H and I from strand 3. The circles represent the positions of the residues when projected onto the surface of the barrel. In this case, θ is the mean twist of the β-sheet about an axis perpendicular to the strand direction. τ is the mean twist of the β-sheet about an axis parallel to the strand direction. ε is the mean coiling of the β-sheet along the strands. η is the mean coiling of the β-sheet along a line perpendicular to the strands. Figure 6C depicts two β-sheets, with the chain direction being shown with arrows. Figure 6D depicts two β-sheets of distance h with angle θ between the average strand vectors. There is also φ, perpendicular to vectors defining θ.

Figures 7A, 7B, 7C and 7D depict four supersecondary structure parameters α/α supersecondary structure parameters for α/α interactions. d is the distance between the helices and θ is the angle between the axes of the helices. σ is defined as the rotation around the helix axis. Ω is the angle between two strand axes after projection onto a plane. In Figures 7C and 7D, the dark circle represents a view of the helix from the end.

Figure 8 depicts the total optimization time vs. value of sorting method for the (a) mixed structural type and (b) β-sheet surface benchmark cases. Sorting is determined by the value of the factor *f* in Eq. 5. The cases exhibit different dependencies on the value of the sorting factor, but both have minima in the vicinity of $f = 0.08$. This trend is observed for all cases (not shown).

Figure 9 depicts the benchmark times of B&T versus other combinatorial search algorithms.

Figure 10 depicts the optimization times resulting from the combination of B&T (hashed bars) and DEE (solid bars) algorithms. The bars on the extreme left and right of the figure are the times for lone B&T and DEE optimization, respectively. The remaining bars are the cumulative B&T and DEE optimization times when the two algorithms are used in succession. The sudden jumps in DEE times arise from lengthy Goldstein doubles calculations.

DETAILED DESCRIPTION OF THE INVENTION

The present invention is directed to the quantitative design and optimization of amino acid sequences, using an "inverse protein folding" approach, which seeks the optimal sequence for a desired structure. Inverse folding is similar to protein design, which seeks to find a sequence or set of sequences that will fold into a desired structure. These approaches can be contrasted with a "protein folding" approach which attempts to predict a structure taken by a given sequence.

The general preferred approach of the present invention is as follows, although alternate embodiments are discussed below. A known protein structure is used as the starting point. The residues to be optimized are then identified, which may be the entire sequence or subset(s) thereof. The side chains of any positions to be varied are then removed. The resulting structure consisting of the protein backbone and the remaining sidechains is called the template. Each variable residue position is then preferably classified as a core residue, a surface residue, or a boundary residue; each classification defines a subset of possible amino acid residues for the position (for example, core residues generally will be selected from the set of hydrophobic residues, surface residues generally will be selected from the hydrophilic residues, and boundary residues may be either). Each amino acid can be represented by a discrete set of all allowed conformers of each side chain, called rotamers. Thus, to arrive at an optimal sequence for a

backbone, all possible sequences of rotamers must be screened, where each backbone position can be occupied either by each amino acid in all its possible rotameric states, or a subset of amino acids, and thus a subset of rotamers.

Two sets of interactions are then calculated for each rotamer at every position: the interaction of the rotamer side chain with all or part of the backbone (the "singles" energy, also called the rotamer/template or rotamer/backbone energy), and the interaction of the rotamer side chain with all other possible rotamers at every other position or a subset of the other positions (the "doubles" energy, also called the rotamer/rotamer energy). The energy of each of these interactions is calculated through the use of a variety of scoring functions, which include the energy of van der Waal's forces, the energy of hydrogen bonding, the energy of secondary structure propensity, the energy of surface area solvation and the electrostatics. Thus, the total energy of each rotamer interaction, both with the backbone and other rotamers, is calculated, and stored in a matrix form. Thus, a sample matrix is generated for the singles calculation, and for the doubles calculations.

The discrete nature of rotamer sets allows a simple calculation of the number of rotamer sequences to be tested. A backbone of length n with m possible rotamers per position will have m^n possible rotamer sequences, a number which grows exponentially with sequence length and renders the calculations either unwieldy or impossible in real time. Accordingly, to solve this combinatorial search problem, either a "Dead End Elimination" (DEE) calculation or a "Branch and Terminate" (B&T) calculation, or both, are performed. The DEE calculation is based on the fact that if the worst total interaction of a first rotamer is still better than the best total interaction of a second rotamer, then the second rotamer cannot be part of the global optimum solution. Since the energies of all rotamers have already been calculated, the DEE approach only requires sums over the sequence length to test and eliminate rotamers, which speeds up the calculations considerably. DEE can be rerun comparing pairs of rotamers, or combinations of rotamers, which will eventually result in the determination of a single sequence which represents the global optimum energy. Alternatively, a B&T calculation can be done, as is more fully described below.

Once the global solution has been found, a Monte Carlo search may be done to generate a rank-ordered list of sequences in the neighborhood of the DEE or B&T solution. Starting at the DEE or B&T solution, random positions are changed to other rotamers, and the new sequence energy is calculated. If the new sequence meets the criteria for acceptance, it is used as a starting point for another jump. After a predetermined number of jumps, a rank-ordered list of sequences is generated.

B&T may also be used to generate a rank ordered list of sequences in the neighborhood of the DEE or B&T solution. In fact, this search may be perfprmed without prior knowledge of the DEE or

B&T solution. The results may then be experimentally verified by physically generating one or more of the protein sequences followed by experimental testing. The information obtained from the testing can then be fed back into the analysis, to modify the procedure if necessary.

Thus, the present invention provides a computer-assisted method of designing a protein. The method comprises providing a protein backbone structure with variable residue positions, and then establishing a group of potential rotamers for each of the residue positions. As used herein, the backbone, or template, includes the backbone atoms and any fixed side chains. The interactions between the protein backbone and the potential rotamers, and between pairs of the potential rotamers, are then processed to generate a set of optimized protein sequences, preferably a single global optimum, which then may be used to generate other related sequences.

Figure 1 illustrates an automated protein design apparatus 20 in accordance with an embodiment of the invention. The apparatus 20 includes a central processing unit 22 which communicates with a memory 24 and a set of input/output devices (e.g., keyboard, mouse, monitor, printer, etc.) 26 through a bus 28. The general interaction between a central processing unit 22, a memory 24, input/output devices 26, and a bus 28 is known in the art. The present invention is directed toward the automated protein design program 30 stored in the memory 24.

The automated protein design program 30 may be implemented with a side chain module 32. As discussed in detail below, the side chain module establishes a group of potential rotamers for a selected protein backbone structure. The protein design program 30 may also be implemented with a ranking module 34. As discussed in detail below, the ranking module 34 analyzes the interaction of rotamers with the protein backbone structure to generate optimized protein sequences. The protein design program 30 may also include a search module 36 to execute a search, for example a Monte Carlo search as described below, in relation to the optimized protein sequences. Finally, an assessment module 38 may also be used to assess physical parameters associated with the derived proteins, as discussed further below.

The memory 24 also stores a protein backbone structure 40, which is downloaded by a user through the input/output devices 26. The memory 24 also stores information on potential rotamers derived by the side chain module 32. In addition, the memory 24 stores protein sequences 44 generated by the ranking module 34. The protein sequences 44 may be passed as output to the input/output devices 26.

The operation of the automated protein design apparatus 20 is more fully appreciated with reference to Fig. 2. Fig. 2 illustrates processing steps executed in accordance with the method of the invention. As described below, many of the processing steps are executed by the protein

design program 30. The first processing step illustrated in Fig. 2 is to provide a protein backbone structure (step 50). As previously indicated, the protein backbone structure is downloaded through the input/output devices 26 using standard techniques.

The protein backbone structure corresponds to a selected protein. By "protein" herein is meant at least two amino acids linked together by a peptide bond. As used herein, protein includes proteins, oligopeptides and peptides. The peptidyl group may comprise naturally occurring amino acids and peptide bonds, or synthetic peptidomimetic structures, i.e. "analogs", such as peptoids (see Simon *et al.*, PNAS USA **89**(20):9367 (1992)).. The amino acids may either be naturally occuring or non-naturally occuring; as will be appreciated by those in the art, any structure for which a set of rotamers is known or can be generated can be used as an amino acid. The side chains may be in either the (R) or the (S) configuration. In a preferred embodiment, the amino acids are in the (S) or (L) configuration.

The chosen protein may be any protein for which a three dimensional structure is known or can be generated; that is, for which there are three dimensional coordinates for each atom of the protein. Generally this can be determined using X-ray crystallographic techniques, NMR techniques, de novo modelling, homology modelling, etc. In general, if X-ray structures are used, structures at 2Å resolution or better are preferred, but not required.

The proteins may be from any organism, including prokaryotes and eukaryotes, with enzymes from bacteria, fungi, extremeophiles such as the archebacteria, insects, fish, animals (particularly mammals and particularly human) and birds all possible.

Suitable proteins include, but are not limited to, industrial and pharmaceutical proteins, including ligands, cell surface receptors, antigens, antibodies, cytokines, hormones, and enzymes. Suitable classes of enzymes include, but are not limited to, hydrolases such as proteases, carbohydrases, lipases; isomerases such as racemases, epimerases, tautomerases, or mutases; transferases, kinases, oxidoreductases, and phophatases. Suitable enzymes are listed in the Swiss-Prot enzyme database.

Suitable protein backbones include, but are not limited to, all of those found in the protein data base compiled and serviced by the Brookhaven National Lab.

Specifically included within "protein" are fragments and domains of known proteins, including functional domains such as enzymatic domains, binding domains, etc., and smaller fragments, such as turns, loops, etc. That is, portions of proteins may be used as well. In addition, protein variants, i.e. non-naturally occuring variants, may be used.

Once the protein is chosen, the protein backbone structure is input into the computer. By "protein backbone structure" or grammatical equivalents herein is meant the three dimensional coordinates that define the three dimensional structure of a particular protein. The structures which comprise a protein backbone structure (of a naturally occuring protein) are the nitrogen, the carbonyl carbon, the α-carbon, and the carbonyl oxygen, along with the direction of the vector from the α-carbon to the β-carbon.

The protein backbone structure which is input into the computer can either include the coordinates for both the backbone and the amino acid side chains, or just the backbone, i.e. with the coordinates for the amino acid side chains removed. If the former is done, the side chain atoms of each amino acid of the protein structure may be "stripped" or removed from the structure of a protein, as is known in the art, leaving only the coordinates for the "backbone" atoms (the nitrogen, carbonyl carbon and oxygen, and the α-carbon, and the hydrogens attached to the nitrogen and α-carbon).

In a preferred embodiment, the protein backbone structure is altered prior to the analysis outlined below. In this embodiment, the representation of the starting protein backbone structure is reduced to a description of the spatial arrangement of its secondary structural elements. The relative positions of the secondary structural elements are defined by a set of parameters called supersecondary structure parameters. These parameters are assigned values that can be systematically or randomly varied to alter the arrangement of the secondary structure elements to introduce explicit backbone flexibility. The atomic coordinates of the backbone are then changed to reflect the altered supersecondary structural parameters, and these new coordinates are input into the system for use in the subsequent protein design automation.

Basically, a protein is first parsed into a collection of secondary structural elements which are then abstracted into geometrical objects. For example, as more fully outlined below, an α-helix is represented by its helical axis and geometric center. The relative orientation and distance between these objects are summarized as super-secondary structure parameters. Concerted backbone motion can be introduced by simply modulating a protein's super-secondary structure parameter values. Accordingly, when all or part of the backbone is to be altered, the portion to be altered is classified as belonging to a particular supersecondary structure element, i.e. α/β, α/α or β/β, and then the supersecondary structural elements as outlined below are altered. As will be appreciated by those in the art, these elements need not be covalently linked, i.e. part of the same protein; for example, this can be done to evaluate protein-protein interactions.

As will be appreciated by those in the art, it is possible to alter the backbone of certain positions, while retaining either a particular amino acid (which can be "floated", as outlined below) or a

particular rotamer at the position; alternatively, both the backbone can be moved and the amino acid side chain can be optimized as outlined herein. Similarly, the backbone can be held constant and only the amino acid side chains are optimized. Combinations of any of these at any position may be done. In general, when supersecondary structural parameters are altered, this is done on more than one amino acid, i.e. the backbone atoms of a plurality of amino acids that contribute to the secondary structure are moved.

As will be appreciated by those in the art, there are a wide variety of different supersecondary structure parameters that can be used. Super-secondary structure parameterization has been described for fold classes that include α/α (Crick FHC The Fourier transform of a coiled-coil. *Acta Crystallogr* 6:685–689 (1953a); Crick FHC. The packing of α-helices. *Acta Crystallogr* 6:689–697 (1953b); Chothia et al., *Proc Natl Acad Sci USA* 78:4146–4150 (1981) "Relative orientation of close-packed β-pleated sheets in proteins" and Chothia et al., *J Mol Biol* 145:215–250 (1981) "Helix to helix packing in proteins"; Chou, et al. Energetics of the structure of the four-α-helix bundle in proteins. *Proc Natl Acad Sci USA* 85:4295–4299 (1988); Murzin AG, Finkelstein AV. General architecture of the α-helical globule. *J Mol Biol* 204:749–769 (1988). Presnell SR, Cohen FE. Topological distribution of four-α-helix bundles. *Proc Natl Acad Sci USA* 86:6592–6596 (1989); Harris et al. Four helix bundle diversity in globular proteins. *J Mol Biol* 236:1356–1368 (1994) , α/β (Chothia et al., Structure of proteins: packing of α-helices and pleated sheets. *Proc Natl Acad Sci USA* 74:4130–4134 (1977); Janin & Chothia, 1980 Packing of α-helices onto β-pleated sheets and the anatomy of *α/β* proteins. *J Mol Biol* 143:95–128; Cohen et al., 1982, Analysis and prediction of the packing of α-helices against a β sheet in the tertiary structure of globular proteins. *J Mol Biol* 156:821–862; Chou et al., 1985, Interactions between an α-helix and β-sheet energetics of *α/β* packing in proteins. *J Mol Biol* 186:591–609, and β/β (Cohen et al., Analysis and prediction of protein β-sheet structures by a combinatorial approach. *Nature* 285:378–382 (1980); Cohen et al., Analysis of the tertiary structure of protein β-sheet sandwiches. *J Mol Biol* 148:253–272 (1981); Chothia & Janin, Relative orientation of close-packed β-pleated sheets in proteins. *Proc Natl Acad Sci USA* 78:4146–4150 (1981); Chothia & Janin, *Proc Natl Acad Sci USA* 78:3955–3965 (1982) Orthogonal packing of β-pleated sheets in proteins; Chou et al., *J Mol Biol* 188:641–649 (1986) "Interactions between two β-sheets energetics of *β/β* packing in proteins"; Laster et al., *Proc Natl Acad Sci USA* 85:3338–3342 (1988)"Structure principles of parallel β-barrels in proteins"; Murzin et al., *J Mol Biol* 236:1369–1381 (1994a), "Principles determining the structure of β-sheet barrels. I. A theoretical analysis"; Murzin et al. *J Mol Biol* 236:1382–1400 (1994b) "Principles determining the structure of β-sheet barrels. II. The observed structures"; all of these references are explicitly incorporated by reference herein in their entirety).

Four different supersecondary structure parameters useful for α/β proteins are shown in Figure 5. In a preferred embodiment, as for all the supersecondary structure parameters, at least one of

these parameter values is altered; other embodiments utilize simultaneous or sequential alteration of two, three or four of these parameter values.

For the α/β protein interactions, the helix center is defined as the average C_α position of the residues chosen for backbone movement. The helix axis is defined as the principal moment of the C_α atoms of these residues (see Chothia et al., 1981, supra). The strand axis is defined as the average of the least-squares lines fit through the midpoints of sequential C_α positions of the two central β-strands. The sheet plane is defined as the least-squares plane fit through the C_α positions of the two central β-strands. The sheet axis is defined as the vector perpendicular to the sheet plane that passes through the helix center. Ω is the angle between the strand axis and the helix axis after projection onto the sheet plane; θ is the angle between the helix axis and the sheet plane; h is the distance between the helix center and the sheet plane; σ is the rotation angle about the helix axis. Backbone alteration requires altering at least one of these parameter values. In a preferred embodiment, the supersecondary structure parameter value Ω is altered by changing the angle degree (either positively or negatively) of up to about 25 degrees, with changes of ± 1°, 2.5°, 5°, 7.5°, and 10° being particularly preferred. In a preferred embodiment, the supersecondary structure parameter value θ is altered by changing the angle degree (either positively or negatively) of up to about 25 degrees, with changes of ± 1°, 2.5°, 5°, 7.5°, and 10° being particularly preferred. the supersecondary structure parameter value σ is altered by changing the angle degree (either positively or negatively) of up to about 25 degrees, with changes of ± 1°, 2.5°, 5°, 7.5°, and 10° being particularly preferred. In a preferred embodiment, the supersecondary structure parameter value h is altered by changes (either positive or negative) of up to about 8 Å, with changes of ± 0.25, 0.50, 0.75, 1.00, 1.25 and 1.5 being particularly preferred. However, as will be appreciated by those in the art, as for all the parameter values outlined herein, larger changes can be made, depending on the protein (i.e. how close or far other secondary structure elements are) and whether other parameter values are made; for example, larger changes in Ω can be made if the helix is also moved away from the sheet (i.e. h is increased).

Four different supersecondary structure parameters useful for α/α proteins are shown in Figure 7. As for α/β parameters, the helix center is defined as the average C_α position of the residues in the helix. The helix axis is defined as the principal moment of the C_α atoms of the residues in the helix. σ is defined as the rotation around the helix axis. Ω is the angle between two strand axes after projection onto a plane. Thus, d, the distance between the helices, can be altered, generally as outlined above for h. Similarly, θ, σ and Ω can be altered as above.

There are a number of different supersecondary structure parameters useful for β/β proteins. β-barrel configurations contain a number of different parameters that can be altered, as shown in Figure 6. These include: (see Figure 6A) R, the barrel radius; α, the angle of tilt of the strands

relative to the barrel axis; and b, the interstrand distance; (see Figure 6B) θ, the mean twist of the β-sheet about an axis perpendicular to the strand direction; τ, the mean twist of the β-sheet about an axis parallel to the strand direction; ε the mean coiling of the β-sheet along the strands; η, the mean coiling of the β-sheet along a line perpendicular to the strands; and (Figure 6C) Ω is angle between the two β-sheet axes. As for the α/β parameter values, each of these may be altered, either positively or negatively. Generally, changes are made in at least one of these parameter values, by changing the angle degree (either positively or negatively) of up to about 25 degrees, with changes of ± 1°, 2.5°, 5°, 7.5°, and 10° being particularly preferred. b can be changed up to ± 1 Å. For β sandwich structures (Figure 6C and 6D), Ω can be altered up to ± 45°, with changes of ± 1°, 2.5°, 5°, 7.5°, and 10° being particularly preferred. Similarly, h can be altered as outlined above for α/β proteins, and θ and ϕ can be altered up to ± 30°.

Once the desired value changes are selected, the coordinate positions for the positions chosen are altered to reflect the change, to form a "new" or "altered" backbone protein structure, i.e. one that has all or part of the backbone atoms in a different position relative to the starting structure. It should be noted that this process can be repeated, i.e. additional backbone changes can be made, on the same or different residues. In addition, after optimization, the backbone of one or more optimal sequences can altered and an optimization can be run.

Alternatively, movement of the backbone can be done manually, i.e. sections of the protein backbone can be randomly or arbitrarily moved. In this embodiment, the backbone atoms of one or more amino acids can be moved some distance, generally an angstrom or more, in any direction. This can be done using standard modeling programs; for example, Molecular Dynamics minimization, Monte Carlo dynamics, or random backbone coordinate/angle motion. It is also possible to move the backbone atoms of single residues, that are either components of secondary structural elements or not.

Once a protein structure backbone is generated (with alterations, as outlined above) and input into the computer, explicit hydrogens are added if not included within the structure (for example, if the structure was generated by X-ray crystallography, hydrogens must be added). After hydrogen addition, energy minimization of the structure is run, to relax the hydrogens as well as the other atoms, bond angles and bond lengths. In a preferred embodiment, this is done by doing a number of steps of conjugate gradient minimization (Mayo *et al.*, J. Phys. Chem. **94**:8897 (1990)) of atomic coordinate positions to minimize the Dreiding force field with no electrostatics. Generally from about 10 to about 250 steps is preferred, with about 50 being most preferred.

The protein backbone structure contains at least one variable residue position. As is known in the art, the residues, or amino acids, of proteins are generally sequentially numbered starting with the

N-terminus of the protein. Thus a protein having a methionine at it's N-terminus is said to have a methionine at residue or amino acid position 1, with the next residues as 2, 3, 4, etc. At each position, the wild type (i.e. naturally occuring) protein may have one of at least 20 amino acids, in any number of rotamers. By "variable residue position" herein is meant an amino acid position of the protein to be designed that is not fixed in the design method as a specific residue or rotamer, generally the wild-type residue or rotamer.

In a preferred embodiment, all of the residue positions of the protein are variable. That is, every amino acid side chain may be altered in the methods of the present invention. This is particularly desirable for smaller proteins, although the present methods allow the design of larger proteins as well. While there is no theoretical limit to the length of the protein which may be designed this way, there is a practical computational limit.

In an alternate preferred embodiment, only some of the residue positions of the protein are variable, and the remainder are "fixed", that is, they are identified in the three dimensional structure as being in a set conformation. In some embodiments, a fixed position is left in its original conformation (which may or may not correlate to a specific rotamer of the rotamer library being used). Alternatively, residues may be fixed as a non-wild type residue; for example, when known site-directed mutagenesis techniques have shown that a particular residue is desirable (for example, to eliminate a proteolytic site or alter the substrate specificity of an enzyme), the residue may be fixed as a particular amino acid. Alternatively, the methods of the present invention may be used to evaluate mutations de novo, as is discussed below. In an alternate preferred embodiment, a fixed position may be "floated"; the amino acid at that position is fixed, but different rotamers of that amino acid are tested. In this embodiment, the variable residues may be at least one, or anywhere from 0.1% to 99.9% of the total number of residues. Thus, for example, it may be possible to change only a few (or one) residues, or most of the residues, with all possibilities in between.

In a preferred embodiment, residues which can be fixed include, but are not limited to, structurally or biologically functional residues. For example, residues which are known to be important for biological activity, such as the residues which form the active site of an enzyme, the substrate binding site of an enzyme, the binding site for a binding partner (ligand/receptor, antigen/antibody, etc.), phosphorylation or glycosylation sites which are crucial to biological function, or structurally important residues, such as disulfide bridges, metal binding sites, critical hydrogen bonding residues, residues critical for backbone conformation such as proline or glycine, residues critical for packing interactions, etc. may all be fixed in a conformation or as a single rotamer, or "floated".

Similarly, residues which may be chosen as variable residues may be those that confer undesirable biological attributes, such as susceptibility to proteolytic degradation, dimerization or aggregation sites, glycosylation sites which may lead to immune responses, unwanted binding activity, unwanted allostery, undesirable enzyme activity but with a preservation of binding, etc.

As will be appreciated by those in the art, the methods of the present invention allow computational testing of "site-directed mutagenesis" targets without actually making the mutants, or prior to making the mutants. That is, quick analysis of sequences in which a small number of residues are changed can be done to evaluate whether a proposed change is desirable. In addition, this may be done on a known protein, or on an protein optimized as described herein.

As will be appreciated by those in the art, a domain of a larger protein may essentially be treated as a small independent protein; that is, a structural or functional domain of a large protein may have minimal interactions with the remainder of the protein and may essentially be treated as if it were autonomous. In this embodiment, all or part of the residues of the domain may be variable.

It should be noted that even if a position is chosen as a variable position, it is possible that the methods of the invention will optimize the sequence in such a way as to select the wild type residue at the variable position. This generally occurs more frequently for core residues, and less regularly for surface residues. In addition, it is possible to fix residues as non-wild type amino acids as well.

Once the protein backbone structure has been selected and input, and the variable residue positions chosen, a group of potential rotamers for each of the variable residue positions is established. This operation is shown as step 52 in Figure 2. This step may be implemented using the side chain module 32. In one embodiment of the invention, the side chain module 32 includes at least one rotamer library, as described below, and program code that correlates the selected protein backbone structure with corresponding information in the rotamer library. Alternatively, the side chain module 32 may be omitted and the potential rotamers 42 for the selected protein backbone structure may be downloaded through the input/output devices 26.

As is known in the art, each amino acid side chain has a set of possible conformers, called rotamers. See Ponder, *et al.*, Acad. Press Inc. (London) Ltd. pp. 775-791 (1987); Dunbrack, *et al.*, Struc. Biol. **1**(5):334-340 (1994); Desmet, *et al.*, Nature **356**:539-542 (1992), all of which are hereby expressly incorporated by reference in their entireity. Thus, a set of discrete rotamers for every amino acid side chain is used. There are two general types of rotamer libraries: backbone dependent and backbone independent. A backbone dependent rotamer library allows different rotamers depending on the position of the residue in the backbone; thus for example, certain leucine rotamers are allowed if the position is within an α helix, and different leucine rotamers are

allowed if the position is not in a α-helix. A backbone independent rotamer library utilizes all rotamers of an amino acid at every position. In general, a backbone independent library is preferred in the consideration of core residues, since flexibility in the core is important. However, backbone independent libraries are computationally more expensive, and thus for surface and boundary positions, a backbone dependent library is preferred. However, either type of library can be used at any position.

In addition, a preferred embodiment does a type of "fine tuning" of the rotamer library by expanding the possible χ (chi) angle values of the rotamers by plus and minus one standard deviation (or more) about the mean value, in order to minimize possible errors that might arise from the discreteness of the library. This is particularly important for aromatic residues, and fairly important for hydrophobic residues, due to the increased requirements for flexibility in the core and the rigidity of aromatic rings; it is not as important for the other residues. Thus a preferred embodiment expands the χ_1 and χ_2 angles for all amino acids except Met, Arg and Lys.

To roughly illustrate the numbers of rotamers, in one version of the Dunbrack & Karplus backbone-dependent rotamer library, alanine has 1 rotamer, glycine has 1 rotamer, arginine has 55 rotamers, threonine has 9 rotamers, lysine has 57 rotamers, glutamic acid has 69 rotamers, asparagine has 54 rotamers, aspartic acid has 27 rotamers, tryptophan has 54 rotamers, tyrosine has 36 rotamers, cysteine has 9 rotamers, glutamine has 69 rotamers, histidine has 54 rotamers, valine has 9 rotamers, isoleucine has 45 rotamers, leucine has 36 rotamers, methionine has 21 rotamers, serine has 9 rotamers, and phenylalanine has 36 rotamers.

In general, proline is not generally used, since it will rarely be chosen for any position, although it can be included if desired. Similarly, a preferred embodiment omits cysteine as a consideration, only to avoid potential disulfide problems, although it can be included if desired.

As will be appreciated by those in the art, other rotamer libraries with all dihedral angles staggered can be used or generated.

In a preferred embodiment, at a minimum, at least one variable position has rotamers from at least two different amino acid side chains; that is, a sequence is being optimized, rather than a structure.

In a preferred embodiment, rotamers from all of the amino acids (or all of them except cysteine, glycine and proline) are used for each variable residue position; that is, the group or set of potential rotamers at each variable position is every possible rotamer of each amino acid. This is especially preferred when the number of variable positions is not high as this type of analysis can be computationally expensive.

In a preferred embodiment, each variable position is classified as either a core, surface or boundary residue position, although in some cases, as explained below, the variable position may be set to glycine to minimize backbone strain.

It should be understood that quantitative protein design or optimization studies prior to the present invention focused almost exclusively on core residues. The present invention, however, provides methods for designing proteins containing core, surface and boundary positions. Alternate embodiments utilize methods for designing proteins containing core and surface residues, core and boundary residues, and surface and boundary residues, as well as core residues alone (using the scoring functions of the present invention), surface residues alone, or boundary residues alone.

The classification of residue positions as core, surface or boundary may be done in several ways, as will be appreciated by those in the art. In a preferred embodiment, the classification is done via a visual scan of the original protein backbone structure, including the side chains, and assigning a classification based on a subjective evaluation of one skilled in the art of protein modelling. Alternatively, a preferred embodiment utilizes an assessment of the orientation of the Cα-Cβ vectors relative to a solvent accessible surface computed using only the template Cα atoms. In a preferred embodiment, the solvent accessible surface for only the Cα atoms of the target fold is generated using the Connolly algorithm with a add-on radius ranging from about 4 to about 12Å, with from about 6 to about 10Å being preferred, and 8 Å being particularly preferred. The Cα radius used ranges from about 1.6Å to about 2.3Å, with from about 1.8 to about 2.1Å being preferred, and 1.95 Å being especially preferred. A residue is classified as a core position if a) the distance for its Cα, along its Cα-Cβ vector, to the solvent accessible surface is greater than about 4-6 Å, with greater than about 5.0 Å being especially preferred, and b) the distance for its Cβ to the nearest surface point is greater than about 1.5-3 Å, with greater than about 2.0 Å being especially preferred. The remaining residues are classified as surface positions if the sum of the distances from their Cα, along their Cα-Cβ vector , to the solvent accessible surface, plus the distance from their Cβ to the closest surface point was less than about 2.5-4 Å, with less than about 2.7 Å being especially preferred. All remaining residues are classified as boundary positions.

Once each variable position is classified as either core, surface or boundary, a set of amino acid side chains, and thus a set of rotamers, is assigned to each position. That is, the set of possible amino acid side chains that the program will allow to be considered at any particular position is chosen. Subsequently, once the possible amino acid side chains are chosen, the set of rotamers that will be evaluated at a particular position can be determined. Thus, a core residue will generally be selected from the group of hydrophobic residues consisting of alanine, valine, isoleucine, leucine, phenylalanine, tyrosine, tryptophan, and methionine (in some embodiments, when the α scaling factor of the van der Waals scoring function, described below, is low, methionine is

removed from the set), and the rotamer set for each core position potentially includes rotamers for these eight amino acid side chains (all the rotamers if a backbone independent library is used, and subsets if a rotamer dependent backbone is used). Similarly, surface positions are generally selected from the group of hydrophilic residues consisting of alanine, serine, threonine, aspartic acid, asparagine, glutamine, glutamic acid, arginine, lysine and histidine. The rotamer set for each surface position thus includes rotamers for these ten residues. Finally, boundary positions are generally chosen from alanine, serine, threonine, aspartic acid, asparagine, glutamine, glutamic acid, arginine, lysine histidine, valine, isoleucine, leucine, phenylalanine, tyrosine, tryptophan, and methionine. The rotamer set for each boundary position thus potentially includes every rotamer for these seventeen residues (assuming cysteine, glycine and proline are not used, although they can be).

Thus, as will be appreciated by those in the art, there is a computational benefit to classifying the residue positions, as it decreases the number of calculations. It should also be noted that there may be situations where the sets of core, boundary and surface residues are altered from those described above; for example, under some circumstances, one or more amino acids is either added or subtracted from the set of allowed amino acids. For example, some proteins which dimerize or multimerize, or have ligand binding sites, may contain hydrophobic surface residues, etc. In addition, residues that do not allow helix "capping" or the favorable interaction with an α-helix dipole may be subtracted from a set of allowed residues. This modification of amino acid groups is done on a residue by residue basis.

In a preferred embodiment, proline, cysteine and glycine are not included in the list of possible amino acid side chains, and thus the rotamers for these side chains are not used. However, in a preferred embodiment, when the variable residue position has a ϕ angle (that is, the dihedral angle defined by 1) the carbonyl carbon of the preceding amino acid; 2) the nitrogen atom of the current residue; 3) the α-carbon of the current residue; and 4) the carbonyl carbon of the current residue) greater than 0°, the position is set to glycine to minimize backbone strain.

Once the group of potential rotamers is assigned for each variable residue position, processing proceeds to step 54 of Figure 2. This processing step entails analyzing interactions of the rotamers with each other and with the protein backbone to generate optimized protein sequences. The ranking module 34 may be used to perform these operations. That is, computer code is written to implement the following functions. Simplistically, as is generally outlined above, the processing initially comprises the use of a number of scoring functions, described below, to calculate energies of interactions of the rotamers, either to the backbone itself or other rotamers.

The scoring functions include a Van der Waals potential scoring function, a hydrogen bond potential scoring function, an atomic solvation scoring function, a secondary structure propensity scoring function and an electrostatic scoring function. As is further described below, at least one scoring function is used to score each position, although the scoring functions may differ depending on the position classification or other considerations, like favorable interaction with an α-helix dipole. As outlined below, the total energy which is used in the calculations is the sum of the energy of each scoring function used at a particular position, as is generally shown in Equation 1:

Equation 1

$$E_{total} = nE_{vdw} + nE_{as} + nE_{h\text{-}bonding} + nE_{ss} + nE_{elec}$$

In Equation 1, the total energy is the sum of the energy of the van der Waals potential (E_{vdw}), the energy of atomic solvation (E_{as}), the energy of hydrogen bonding ($E_{h\text{-}bonding}$), the energy of secondary structure (E_{ss}) and the energy of electrostatic interaction (E_{elec}). The term n is either 0 or 1, depending on whether the term is to be considered for the particular residue position, as is more fully outlined below.

In a preferred embodiment, a van der Waals' scoring function is used. As is known in the art, van der Waals' forces are the weak, non-covalent and non-ionic forces between atoms and molecules, that is, the induced dipole and electron repulsion (Pauli principle) forces.

The van der Waals scoring function is based on a van der Waals potential energy. There are a number of van der Waals potential energy calculations, including a Lennard-Jones 12/6 potential with radii and well depth parameters from the Dreiding force field, Mayo *et al.*, J. Prot. Chem., 1990, expressly incorporated herein by reference, or the exponential 6 potential. Equation 2, shown below, is the preferred Lennard-Jones potential:

Equation 2

$$E_{vdw} = D_0\left\{\left(\frac{R_0}{R}\right)^{12} - 2\left(\frac{R_0}{R}\right)^{6}\right\}$$

R_0 is the geometric mean of the van der Waals radii of the two atoms under consideration, and D_0 is the geometric mean of the well depth of the two atoms under consideration. E_{vdw} and R are the energy and interatomic distance between the two atoms under consideration, as is more fully described below.

In a preferred embodiment, the van der Waals forces are scaled using a scaling factor, α. Equation 3 shows the use of α in the van der Waals Lennard-Jones potential equation:

Equation 3

$$E_{vdw} = D_0\left\{\left(\frac{\alpha R_0}{R}\right)^{12} - 2\left(\frac{\alpha R_0}{R}\right)^{6}\right\}$$

The role of the α scaling factor is to change the importance of packing effects in the optimization and design of any particular protein. Specifically, a reduced van der Waals steric constraint can compensate for the restrictive effect of a fixed backbone and discrete side-chain rotamers in the simulation and can allow a broader sampling of sequences compatible with a desired fold. In a preferred embodiment, α values ranging from about 0.70 to about 1.10 can be used, with α values from about 0.8 to about 1.05 being preferred, and from about 0.85 to about 1.0 being especially preferred. Specific α values which are preferred are 0.80, 0.85, 0.90, 0.95, 1.00, and 1.05.

Generally speaking, variation of the van der Waals scale factor α results in four regimes of packing specificity: regime 1 where $0.9 \leq \alpha \leq 1.05$ and packing constraints dominate the sequence selection; regime 2 where $0.8 \leq \alpha < 0.9$ and the hydrophobic solvation potential begins to compete with packing forces; regime 3 where $\alpha < 0.8$ and hydrophobic solvation dominates the design; and, regime 4 where $\alpha > 1.05$ and van der Waals repulsions appear to be too severe to allow meaningful sequence selection. In particular, different α values may be used for core, surface and boundary positions, with regimes 1 and 2 being preferred for core residues, regime 1 being preferred for surface residues, and regime 1 and 2 being preferred for boundary residues.

In a preferred embodiment, the van der Waals scaling factor is used in the total energy calculations for each variable residue position, including core, surface and boundary positions.

In a preferred embodiment, an atomic solvation potential scoring function is used. As is appreciated by those in the art, solvent interactions of a protein are a significant factor in protein stability, and residue/protein hydrophobicity has been shown to be the major driving force in protein folding. Thus, there is an entropic cost to solvating hydrophobic surfaces, in addition to the potential for misfolding or aggregation. Accordingly, the burial of hydrophobic surfaces within a protein structure is beneficial to both folding and stability. Similarly, there can be a disadvantage for burying hydrophilic residues. The accessible surface area of a protein atom is generally defined as the area of the surface over which a water molecule can be placed while making van der Waals contact with this atom and not penetrating any other protein atom. Thus, in a preferred embodiment, the solvation potential is generally scored by taking the total possible exposed surface area of the moiety or two independent moieties (either a rotamer or the first rotamer and the second rotamer), which is the reference, and subtracting out the "buried" area, i.e. the area

which is not solvent exposed due to interactions either with the backbone or with other rotamers. This thus gives the exposed surface area.

Alternatively, a preferred embodiment calculates the scoring function on the basis of the "buried" portion; i.e. the total possible exposed surface area is calculated, and then the calculated surface area after the interaction of the moieties is subtracted, leaving the buried surface area. A particularly preferred method does both of these calculations.

As is more fully described below, both of these methods can be done in a variety of ways. See Eisenberg *et al.*, Nature **319**:199-203 (1986); Connolly, Science **221**:709-713 (1983); and Wodak, *et al.*, Proc. Natl. Acad. Sci. USA **77**(4):1736-1740 (1980), all of which are expressly incorporated herein by reference. As will be appreciated by those in the art, this solvation potential scoring function is conformation dependent, rather than conformation independent.

In a preferred embodiment, the pairwise solvation potential is implemented in two components, "singles" (rotamer/template) and "doubles" (rotamer/rotamer), as is more fully described below. For the rotamer/template buried area, the reference state is defined as the rotamer in question at residue position i with the backbone atoms only of residues i-1, i and i+1, although in some instances just i may be used. Thus, in a preferred embodiment, the solvation potential is not calculated for the interaction of each backbone atom with a particular rotamer, although more may be done as required. The area of the side chain is calculated with the backbone atoms excluding solvent but not counted in the area. The folded state is defined as the area of the rotamer in question at residue i, but now in the context of the entire template structure including non-optimized side chains, i.e. every other fixed position residue. The rotamer/template buried area is the difference between the reference and the folded states. The rotamer/rotamer reference area can be done in two ways; one by using simply the sum of the areas of the isolated rotamers; the second includes the full backbone. The folded state is the area of the two rotamers placed in their relative positions on the protein scaffold but with no template atoms present. In a preferred embodiment, the Richards definition of solvent accessible surface area (Lee and Richards, J. Mol. Biol. 55:379-400, 1971, hereby incorporated by reference) is used, with a probe radius ranging from 0.8 to 1.6 Å, with 1.4 Å being preferred, and Drieding van der Waals radii, scaled from 0.8 to 1.0. Carbon and sulfur, and all attached hydrogens, are considered nonpolar. Nitrogen and oxygen, and all attached hydrogens, are considered polar. Surface areas are calculated with the Connolly algorithm using a dot density of 10 Å-2 (Connolly, (1983) (supra), hereby incorporated by reference).

In a preferred embodiment, there is a correction for a possible overestimation of buried surface area which may exist in the calculation of the energy of interaction between two rotamers (but not

the interaction of a rotamer with the backbone). Since, as is generally outlined below, rotamers are only considered in pairs, that is, a first rotamer is only compared to a second rotamer during the "doubles" calculations, this may overestimate the amount of buried surface area in locations where more than two rotamers interact, that is, where rotamers from three or more residue positions come together. Thus, a correction or scaling factor is used as outlined below.

The general energy of solvation is shown in Equation 4:

Equation 4

$$E_{sa} = f(SA)$$

where E_{sa} is the energy of solvation, f is a constant used to correlate surface area and energy, and SA is the surface area. This equation can be broken down, depending on which parameter is being evaluated. Thus, when the hydrophobic buried surface area is used, Equation 5 is appropriate:

Equation 5

$$E_{sa} = f_1(SA_{buried\ hydrophobic})$$

where f_1 is a constant which ranges from about 10 to about 50 cal/mol/Å^2, with 23 or 26 cal/mol/Å^2 being preferred. When a penalty for hydrophilic burial is being considered, the equation is shown in Equation 6:

Equation 6

$$E_{sa} = f_1(SA_{buried\ hydrophobic}) + f_2(SA_{buried\ hydrophilic})$$

where f_2 is a constant which ranges from -50 to -250 cal/mol/Å^2, with -86 or -100 cal/mol/Å^2 being preferred. Similarly, if a penalty for hydrophobic exposure is used, equation 7 or 8 may be used:

Equation 7

$$E_{sa} = f_1(SA_{buried\ hydrophobic}) + f_3(SA_{exposed\ hydrophobic})$$

Equation 8

$$E_{sa} = f_1(SA_{buried\ hydrophobic}) + f_2(SA_{buried\ hydrophilic}) + f_3(SA_{exposed\ hydrophobic}) + f_4(SA_{exposed\ hydrophilic})$$

In a preferred embodiment, $f_3 = -f_1$.

In one embodiment, backbone atoms are not included in the calculation of surface areas, and values of 23 cal/mol/Å^2 (f_1) and -86 cal/mol/Å^2 (f_2) are determined.

In a preferred embodiment, this overcounting problem is addressed using a scaling factor that compensates for only the portion of the expression for pairwise area that is subject to over-counting. In this embodiment, values of -26 cal/mol/Å^2 (f_1) and 100 cal/mol/Å^2 (f_2) are determined.

Atomic solvation energy is expensive, in terms of computational time and resources. Accordingly, in a preferred embodiment, the solvation energy is calculated for core and/or boundary residues, but not surface residues, with both a calculation for core and boundary residues being preferred, although any combination of the three is possible.

In a preferred embodiment, a hydrogen bond potential scoring function is used. A hydrogen bond potential is used as predicted hydrogen bonds do contribute to designed protein stability (see Stickle *et al.*, J. Mol. Biol. 226:1143 (1992); Huyghues-Despointes *et al.*, Biochem. 34:13267 (1995), both of which are expressly incorporated herein by reference). As outlined previously, explicit hydrogens are generated on the protein backbone structure.

In a preferred embodiment, the hydrogen bond potential consists of a distance-dependent term and an angle-dependent term, as shown in Equation 9:

Equation 9

$$E_{H\text{-}Bonding} = D_0\left\{5\left(\frac{R_0}{R}\right)^{12} - 6\left(\frac{R_0}{R}\right)^{10}\right\}F(\theta,\phi,\varphi)$$

where R_0 (2.8 Å) and D_0 (8 kcal/mol) are the hydrogen-bond equilibrium distance and well-depth, respectively, and R is the donor to acceptor distance. This hydrogen bond potential is based on the potential used in DREIDING with more restrictive angle-dependent terms to limit the occurrence of unfavorable hydrogen bond geometries. The angle term varies depending on the hybridization state of the donor and acceptor, as shown in Equations 10, 11, 12 and 13. Equation 10 is used for sp^3 donor to sp^3 acceptor; Equation 11 is used for sp^3 donor to sp^2 acceptor, Equation 12 is used for sp^2 donor to sp^3 acceptor, and Equation 13 is used for sp^2 donor to sp^2 acceptor:

Equation 10

$$F = \cos^2\theta\cos^2(\phi - 109.5)$$

Equation 11

$$F = \cos^2\theta\cos^2\phi$$

Equation 12

$$F = \cos^4\theta$$

Equation 13

$$F = \cos^2\theta \cos^2(\max[\phi, \varphi])$$

In Equations 10-13, θ is the donor-hydrogen-acceptor angle, ϕ is the hydrogen-acceptor-base angle (the base is the atom attached to the acceptor, for example the carbonyl carbon is the base for a carbonyl oxygen acceptor), and φ is the angle between the normals of the planes defined by the six atoms attached to the sp^2 centers (the supplement of φ is used when φ is less than 90°). The hydrogen-bond function is only evaluated when 2.6 Å ≤ R ≤ 3.2 Å, θ > 90°, ϕ - 109.5° < 90° for the sp^3 donor - sp^3 acceptor case, and, ϕ > 90° for the sp^3 donor - sp^2 acceptor case; preferably, no switching functions are used. Template donors and acceptors that are involved in template-template hydrogen bonds are preferably not included in the donor and acceptor lists. For the purpose of exclusion, a template-template hydrogen bond is considered to exist when 2.5 Å ≤ R ≤ 3.3 Å and θ ≥ 135°.

The hydrogen-bond potential may also be combined or used with a weak coulombic term that includes a distance-dependent dielectric constant of 40R, where R is the interatomic distance. Partial atomic charges are preferably only applied to polar functional groups. A net formal charge of +1 is used for Arg and Lys and a net formal charge of -1 is used for Asp and Glu; see Gasteiger, *et al.*, Tetrahedron **36**:3219-3288 (1980); Rappe, *et al.*, J. Phys. Chem. **95**:3358-3363 (1991).

In a preferred embodiment, an explicit penalty is given for buried polar hydrogen atoms which are not hydrogen bonded to another atom. See Eisenberg, *et al.*, (1986) (supra), hereby expressly incorporated by reference. In a preferred embodiment, this penalty for polar hydrogen burial, is from about 0 to about 3 kcal/mol, with from about 1 to about 3 being preferred and 2 kcal/mol being particularly preferred. This penalty is only applied to buried polar hydrogens not involved in hydrogen bonds. A hydrogen bond is considered to exist when E_{HB} ranges from about 1 to about 4 kcal/mol, with E_{HB} of less than -2 kcal/mol being preferred. In addition, in a preferred embodiment, the penalty is not applied to template hydrogens, i.e. unpaired buried hydrogens of the backbone.

In a preferred embodiment, only hydrogen bonds between a first rotamer and the backbone are scored, and rotamer-rotamer hydrogen bonds are not scored. In an alternative embodiment, hydrogen bonds between a first rotamer and the backbone are scored, and rotamer-rotamer hydrogen bonds are scaled by 0.5 .

In a preferred embodiment, the hydrogen bonding scoring function is used for all positions, including core, surface and boundary positions. In alternate embodiments, the hydrogen bonding scoring function may be used on only one or two of these.

In a preferred embodiment, a secondary structure propensity scoring function is used. This is based on the specific amino acid side chain, and is conformation independent. That is, each amino acid has a certain propensity to take on a secondary structure, either α-helix or β-sheet, based on its ϕ and ψ angles. See Muñoz *et al.*, Current Op. in Biotech. **6**:382 (1995); Minor, *et al.*, Nature **367**:660-663 (1994); Padmanabhan, *et al.*, Nature **344**:268-270 (1990); Muñoz, *et al.*, Folding & Design **1**(3):167-178 (1996); and Chakrabartty, *et al.*, Protein Sci. 3:843 (1994), all of which are expressly incorporated herein by reference. Thus, for variable residue positions that are in recognizable secondary structure in the backbone, a secondary structure propensity scoring function is preferably used. That is, when a variable residue position is in an α-helical area of the backbone, the α-helical propensity scoring function described below is calculated. Whether or not a position is in a α-helical area of the backbone is determined as will be appreciated by those in the art, generally on the basis of ϕ and ψ angles; for α-helix, ϕ angles from -2 to -70 and ψ angles from -30 to -100 generally describe an α-helical area of the backbone.

Similarly, when a variable residue position is in a β-sheet backbone conformation, the β-sheet propensity scoring function is used. β-sheet backbone conformation is generally described by ϕ angles from -30 to -100 and χ angles from +40 to +180. In alternate preferred embodiments, variable residue positions which are within areas of the backbone which are not assignable to either β-sheet or α-helix structure may also be subjected to secondary structure propensity calculations.

In a preferred embodiment, energies associated with secondary propensities are calculated using Equation 14.

Equation 14

$$E_{\alpha} = 10^{N_{ss}(\Delta G^{\circ}_{aa} - \Delta G^{\circ}_{ala})} - 1$$

In Equation 14, E_α (or E_β) is the energy of α-helical propensity, ΔG°_{aa} is the standard free energy of helix propagation of the amino acid, and ΔG°_{ala} is the standard free energy of helix propagation of alanine used as a standard, or standard free energy of β-sheet formation of the amino acid, both of which are available in the literature (see Chakrabartty, *et al.*, (1994) (supra), and Munoz, *et al.*, (1996) (supra)), both of which are expressly incorporated herein by reference), and N_{ss} is the propensity scale factor which is set to range from 1 to 4, with 2.0 being preferred. This potential is preferably selected in order to scale the propensity energies to a similar range as the other terms in the scoring function.

In a preferred embodiment, β-sheet propensities are preferably calculated only where the i-1 and i+1 residues are also in β-sheet conformation.

In a preferred embodiment, the secondary structure propensity scoring function is used only in the energy calculations for surface variable residue positions. In alternate embodiments, the secondary structure propensity scoring function is used in the calculations for core and boundary regions as well.

In a preferred embodiment, an electrostatic scoring function is used, as shown below in Equation 15:

Equation 15

$$E_{elec} = \frac{qq'}{\epsilon r^2}$$

In this Equation, q is the charge on atom 1, q' is charge on atom 2, and r is the interaction distance.

In a preferred embodiment, at least one scoring function is used for each variable residue position; in preferred embodiments, two, three or four scoring functions are used for each variable residue position.

Once the scoring functions to be used are identified for each variable position, the preferred first step in the computational analysis comprises the determination of the interaction of each possible rotamer with all or part of the remainder of the protein. That is, the energy of interaction, as measured by one or more of the scoring functions, of each possible rotamer at each variable residue position with either the backbone or other rotamers, is calculated. In a preferred embodiment, the interaction of each rotamer with the entire remainder of the protein, i.e. both the entire template and all other rotamers, is done. However, as outlined above, it is possible to only model a portion of a protein, for example a domain of a larger protein, and thus in some cases, not all of the protein need be considered.

In a preferred embodiment, the first step of the computational processing is done by calculating two sets of interactions for each rotamer at every position (step 70 of figure 3): the interaction of the rotamer side chain with the template or backbone (the "singles" energy), and the interaction of the rotamer side chain with all other possible rotamers at every other position (the "doubles" energy), whether that position is varied or floated. It should be understood that the backbone in this case includes both the atoms of the protein structure backbone, as well as the atoms of any fixed residues, wherein the fixed residues are defined as a particular conformation of an amino acid.

Thus, "singles" (rotamer/template) energies are calculated for the interaction of every possible rotamer at every variable residue position with the backbone, using some or all of the scoring functions. Thus, for the hydrogen bonding scoring function, every hydrogen bonding atom of the rotamer and every hydrogen bonding atom of the backbone is evaluated, and the E_{HB} is calculated for each possible rotamer at every variable position. Similarly, for the van der Waals scoring

function, every atom of the rotamer is compared to every atom of the template (generally excluding the backbone atoms of its own residue), and the E_{vdw} is calculated for each possible rotamer at every variable residue position. In addition, generally no van der Waals energy is calculated if the atoms are connected by three bonds or less. For the atomic solvation scoring function, the surface of the rotamer is measured against the surface of the template, and the E_{as} for each possible rotamer at every variable residue position is calculated. The secondary structure propensity scoring function is also considered as a singles energy, and thus the total singles energy may contain an E_{ss} term. As will be appreciated by those in the art, many of these energy terms will be close to zero, depending on the physical distance between the rotamer and the template position; that is, the farther apart the two moieties, the closer to zero.

Accordingly, as outlined above, the total singles energy is the sum of the energy of each scoring function used at a particular position, as shown in Equation 1, wherein n is either 1 or zero, depending on whether that particular scoring function was used at the rotamer position:

Equation 1

$$E_{total} = nE_{vdw} + nE_{as} + nE_{h\text{-}bonding} + nE_{ss} + nE_{elec}$$

Once calculated, each singles E_{total} for each possible rotamer is stored in the memory 24 within the computer, such that it may be used in subsequent calculations, as outlined below.

For the calculation of "doubles" energy (rotamer/rotamer), the interaction energy of each possible rotamer is compared with every possible rotamer at all other variable residue positions. Thus, "doubles" energies are calculated for the interaction of every possible rotamer at every variable residue position with every possible rotamer at every other variable residue position, using some or all of the scoring functions. Thus, for the hydrogen bonding scoring function, every hydrogen bonding atom of the first rotamer and every hydrogen bonding atom of every possible second rotamer is evaluated, and the E_{HB} is calculated for each possible rotamer pair for any two variable positions. Similarly, for the van der Waals scoring function, every atom of the first rotamer is compared to every atom of every possible second rotamer, and the E_{vdW} is calculated for each possible rotamer pair at every two variable residue positions. For the atomic solvation scoring function, the surface of the first rotamer is measured against the surface of every possible second rotamer, and the E_{as} for each possible rotamer pair at every two variable residue positions is calculated. The secondary structure propensity scoring function need not be run as a "doubles" energy, as it is considered as a component of the "singles" energy. As will be appreciated by those in the art, many of these double energy terms will be close to zero, depending on the physical distance between the first rotamer and the second rotamer; that is, the farther apart the two moieties, the closer to zero.

Accordingly, as outlined above, the total doubles energy is the sum of the energy of each scoring function used to evaluate every possible pair of rotamers, as shown in Equation 16, wherein n is either 1 or zero, depending on whether that particular scoring function was used at the rotamer position:

Equation 18

$$E_{total} = nE_{vdw} + nE_{as} + nE_{h\text{-}bonding} + E_{elec}$$

An example is illuminating. A first variable position, i, has three (an unrealistically low number) possible rotamers (which may be either from a single amino acid or different amino acids) which are labelled i_a, i_b, and i_c. A second variable position, j, also has three possible rotamers, labelled j_d, j_e, and j_f. Thus, nine doubles energies (E_{total}) are calculated in all: $E_{total}(i_a, j_d)$, $E_{total}(i_a, j_e)$, $E_{total}(i_a, j_f)$, $E_{total}(i_b, j_d)$, $E_{total}(i_b, j_e)$, $E_{total}(i_b, j_f)$, $E_{total}(i_c, j_d)$, $E_{total}(i_c, j_e)$, and $E_{total}(i_c, j_f)$.

Once calculated, each doubles E_{total} for each possible rotamer pair is stored in memory 24 within the computer, such that it may be used in subsequent calculations, as outlined below.

Once the singles and doubles energies are calculated and stored, the next step of the computational processing may occur. Generally speaking, the goal of the computational processing is to determine a set of optimized protein sequences. By "optimized protein sequence" herein is meant a sequence that best fits the mathematical equations herein. As will be appreciated by those in the art, a global optimized sequence is the one sequence that best fits Equation 1, i.e. the sequence that has the lowest energy of any possible sequence. However, there are any number of sequences that are not the global minimum but that have low energies.

In a preferred embodiment, the set comprises the globally optimal sequence in its optimal conformation, i.e. the optimum rotamer at each variable position. That is, computational processing is run until the simulation program converges on a single sequence which is the global optimum.

In a preferred embodiment, the set comprises at least two optimized protein sequences. Thus for example, the computational processing step may eliminate a number of disfavored combinations but be stopped prior to convergence, providing a set of sequences of which the global optimum is one. In addition, further computational analysis, for example using a different method, may be run on the set, to further eliminate sequences or rank them differently. Alternatively, as is more fully described below, the global optimum may be reached, and then further computational processing may occur, which generates additional optimized sequences in the neighborhood of the global optimum.

If a set comprising more than one optimized protein sequences is generated, they may be rank ordered in terms of theoretical quantitative stability, as is more fully described below.

In a preferred embodiment, the computational processing step first comprises an elimination step, sometimes referred to as "applying a cutoff", either a singles elimination or a doubles elimination. Singles elimination comprises the elimination of all rotamers with template interaction energies of greater than about 10 kcal/mol prior to any computation, with elimination energies of greater than about 15 kcal/mol being preferred and greater than about 25 kcal/mol being especially preferred. Similarly, doubles elimination is done when a rotamer has interaction energies greater than about 10 kcal/mol with all rotamers at a second residue position, with energies greater than about 15 being preferred and greater than about 25 kcal/mol being especially preferred.

In a preferred embodiment, the computational processing comprises direct determination of total sequence energies, followed by comparison of the total sequence energies to ascertain the global optimum and rank order the other possible sequences, if desired. The energy of a total sequence is shown below in Equation 17:

Equation 17

$$E_{\text{total protein}} = E_{(b-b)} + \sum_{\text{all } i} E_{(i_a)} + \sum_{\text{all } i} \sum_{\text{all } j \text{ pairs}} E_{(i_a,j_a)}$$

Thus every possible combination of rotamers may be directly evaluated by adding the backbone-backbone (sometimes referred to herein as template-template) energy ($E_{(b-b)}$ which is constant over all sequences herein since the backbone is kept constant), the singles energy for each rotamer (which has already been calculated and stored), and the doubles energy for each rotamer pair (which has already been calculated and stored). Each total sequence energy of each possible rotamer sequence can then be ranked, either from best to worst or worst to best. This is obviously computationally expensive and becomes unwieldy as the length of the protein increases.

Thus, as outlined herein, the computational processing includes one or more Branched & Terminated (B&T) computational steps as outlined below, and optionally a DEE step, also outlined below.

Accordingly, the present invention provides a novel deterministic combinatorial search algorithm, called "Branch and Terminate" (B&T) derived from the Branch-and-Bound search method. The B&T approach is based on the construction of an efficient, but very restrictive bounding expression, which is used for the search of a combinatorial tree representing the protein system. The bounding expression is used both to determine the optimal organization of the tree and to perform a highly effective pruning procedure named "termination." For some calculations, the B&T method rivals

the current deterministic standard, Dead-End Elimination (DEE), sometimes finding the solution up to 21 times faster. A more significant feature of B&T algorithm is that it can provide an efficient way to complete the optimization of problems that have been partially reduced by a DEE algorithm.

The B&T algorithm is an effective optimization algorithm when used alone. Moreover, it can increase the problem size limit of amino acid side chain placement calculations, such as protein design, by completing DEE optimizations that reach a point at which the DEE criteria become inefficient. Together the two algorithms make it possible to find solutions to problems that are intractable by either algorithm alone.

In a preferred embodiment, B&T is used when DEE algorithms are not sufficient, due either to the nature of their energy distributions or their sheer size. For example, the optimization of long hydrophilic side chains on β-sheets is typically composed of large numbers of rotamers with interaction energies that are very small in magnitude. DEE is able to reduce the combinatorial size of the problem significantly at the outset, but soon after, elimination becomes inefficient, relying entirely on computationally expensive DEE doubles calculations (Lasters, I. & Desmet, J., *Prot. Eng.* **6**, 717-722 (1993); Gordon, D.B. & Mayo, S.L., *J. Comp. Chem.* **19**, 1505-1514 (1998)). This behavior is also observed in the later stages of very large calculations, when after several rounds of unification further eliminations become difficult and the number of super-rotamers at super-residue positions becomes very large (Desmet, J., et al., In *The Protein Folding Problem and Tertiary Structure Prediction*. Merz Jr., K. & Le Grand, S. Ed., Birkhäuser, Boston. p. 307 (1994)). To complete such calculations, a technique consisting of exhaustive combinatorial build-up aided by DEE has been described (De Maeyer, M., et al., *Folding & Design*, **2**, 53-56 (1997)). However, since the effectiveness of the elimination criteria is poor in these cases, it is advantageous to construct a method that is not dependent on them.

To address these difficult optimization problems, the invention provides "Branch-and-Terminate" (B&T) methods, based on a "Branched & Bound" (B&B) algorithm. B&B algorithms comprise a sub-class of backtrack algorithms that utilize information about costs (or energies) of complete and partial solutions. Backtrack algorithms are commonly used in atomic-level simulations to construct self-avoiding chains, and they have been used in protein design to engineer metal binding sites into proteins (Hellinga, H.W. & Richards, F.M., *J. Mol. Biol.* **222**, 763-785 (1991)).

B&B algorithms are commonly applied to theoretical combinatorial and scheduling problems, and more recently to combinatorial problems of structural biology ranging from sequence alignment (Lathrop, R.H. & Smith, T.F., *J. Mol. Biol.* **255**, 641-665 (1996)), and structural comparison (Escalier, V., et al., *J Comp. Biol.* **5**, 41-56 (1998)), to macromolecular packing (Wang, C.S.E., et al., *Proteins,* **32**, 26-42 (1998)), ligand design (Todorov, N.P. & Dean, P.M., *J. Comp. Aid. Mol.*

Des. **12**, 335-349 (1998)), and recently, protein tertiary structure prediction (Eyrich, V. A. et al., *Proteins.* **35**, 41-57 (1999)). Toward the study of protein side-chains, Samudrala and Moult (Samudrala, R. & Moult, J., *J. Mol. Biol.* **279**, 287-302 (1998)) have described a graph-theoretic approach to the closely related problem of comparative modeling, in which they represent the search as a clique-finding problem, which they solve using a B&B algorithm. In addition, Leach and Lemon (Leach, A.R. & Lemon, A.P., *Proteins.* **33**, 227-239 (1998)) have used a B&B algorithm (called "A*") to explore the conformational energy surface of protein side-chains.

It is straightforward to formulate the side chain optimization problem for direct optimization by a B&B algorithm. All that is necessary is to describe the problem as a search of a combinatorial tree where one searches for the single path through the branches that corresponds to the global minimum energy conformation (GMEC) set of rotamers. The B&B algorithm is effective because it simultaneously prunes the tree while searching; each branch is tested with a quantitative bounding expression before being searched. The addition of "termination" functions as described herein increases the optimization speed dramatically. The description of the B&T algorithm that follows is tailored for rotamer selection, but the algorithm is in fact generalizable to any combinatorial optimziation problem in which all the interaction energies are pairwise and pre-computable. The bounding expression is similarly general.

First, a bounding function is used that maximizes the efficiency of pruning for problems in which to total energy can be decomposed into interactions between pairs of rotamers. When a combinatorial tree is used to describe the side chain optimization problem, the root of the tree is placed at the top, and branches extend downward. Each level of depth of the tree corresponds to an amino acid position, and each node represents a particular rotamer choice at that position. Thus a path that extends all the way from the tree root through all levels of branches to a leaf describes a complete rotamer sequence. The problem, then, is to search for the path corresponding to the sequence with the lowest energy.

A partial path from the root describes a rotamer sequence that is incompletely specified. Alternatively, the path can be interpreted physically as specifying a unique composite rotamer, or "super-rotamer" that occupies a subset of the amino acid positions. Extending the path deeper into the tree corresponds to appending additional rotamers to the super-rotamer, which can be repeated until all positions are specified. According to this interpretation, a full search of the tree would entail the construction of all possible super-rotamers to completion.

It is often possible, however, to determine that a particular partially-specified super-rotamer is not part of the GMEC. In such a case, it is unnecessary to explore any combinations that would result from building up the super-rotamer further. Applied recursively, such observations prune sub-trees

from nodes throughout the tree, thereby enabling an exhaustive search without complete enumeration of all possible super-rotamers.

The pruning determination is accomplished by comparing a lower energy bound for the partially-specified rotamer sequence to a known reference energy. As shown in Equation 18, given a reference energy of any plausible sequence, it must be true that the energy of the GMEC is less than or equal to the energy of any plausible sequence.

Equation 18

$$E_{GMEC} \leq E_{reference}$$

One may therefore deduce that the global minimum does not contain a particular super-rotamer upon observing as in Equation 19 that the energy $E_{super,best}$ of the sequence resulting from optimal completion of the candidate super-rotamer is greater than the reference energy.

Equation 19

$$E_{super,\ best} > E_{reference}$$

Finding the optimal completing sequence, however, can be as difficult as the original problem, so we instead construct an expression for a lower energy bound, $E_{super,bound}$. The expression is constructed to compute an inexpensive lower energy bound based on the partially specified sequence, as well as on the rotamers that are available at the unspecified positions. By definition, the bound must satisfy the inequality,

Equation 20

$$E_{super\ best} \geq E_{super,\ bound}$$

With this quantity in hand, we may prune any sub-tree for which we observe that the lower bound is greater than the reference energy.

Equation 21

$$E_{super,\ bound} > E_{reference}$$

This is the bounding criterion. The Branch-and-Bound algorithm consists of an exhaustive traversal of the combinatorial tree, applying this criterion to each node as it is encountered.

Whenever the search produces a complete path with an energy lower than the current reference energy, the reference energy is updated. This way, the effectiveness of the bounding criterion is increased over the course of the optimization. Moreover, upon completion of the search, the reference energy is the global minimum energy. The corresponding sequence is also stored during each update, which produces the corresponding GMEC.

The successful implementation of a B&B type of algorithm depends largely on the construction of the bounding expression. A bounding expression that is very stringent will produce lower bounds that are high in energy, and therefore will result in more sub-trees that can be pruned by the bounding criterion. The size of the resulting tree will be smaller than one pruned by a less stringent expression, and the search will be faster. It is therefore important to design the bounding expression to most fully utilize the sequence information available.

On the other hand, stringency is obtained at the cost of time. A maximally stringent bound might prune all sub-trees except for the one containing the global minimum, but it would take an impractical amount of time to compute. It is therefore also necessary to temper stringency with speed considerations in order to obtain a bounding expression that is properly balanced for efficient searching.

The construction of such a bounding expression is shown in Example 1. Given a partially constructed super-rotamer and the available rotamers at the remaining positions, the approach is to utilize the corresponding energetic information as fully as possible while keeping the computational order of the bounding expression constant. The result is a novel, highly-effective bounding expression that provides the basis for the remaining B&T techniques.

An additional advantage of the B&T allogrithm is the form of the resulting expression; it isolates those parts of the expression that are identical for rotamers on the same level of a sub-tree. Thus it is possible to further increase the efficiency of the search by precomputing these shared quantities as each group of nodes is encountered, rather than redundantly evaluating the entire bounding expression for every unique node. This method is described in Example 1.

In addition to the bounding function, the invention further provides a termination function, in which the bounding function is used to deterministically remove rotamers at all amino acid positions, thereby reducing the overall size of the tree before searching. Termination is additionally effective when performed at every level of recursion of the search, sometimes increasing the overall speed of the optimization by an order of magnitude.

The enhancements of the B&T algorithm relative to the B&B method are based on a process called "termination." Because all the pairwise interactions are precomputable, the organization of the combinatorial tree is arbitrary (i.e. there is no specific order to which different amino acid positions must be assigned to different levels of the tree). However, organization of the tree can have a significant influence on the speed of the calculation. For example, a greater reduction in the size of

the search is derived from pruning a branch at the root of the tree rather than pruning a branch closer to the leaves. Placing a branch at the leaves that would be pruned if placed at the root would be inefficient because the same pruning step would necessarily be repeated for every leaf.

In fact, it commonly occurs that all amino acid positions have some rotamers that could be pruned if placed at the root of the tree. To circumvent the potential loss of efficiency, we implement a pre-processing procedure before determining the tree organization. This procedure consists of temporarily considering each amino acid position to be at the root level and checking if any of its rotamers can be immediately pruned. All rotamers pruned from root positions may be completely discarded for the remainder of the optimization, and are dubbed "terminated" to reflect this fact. The result is an overall reduction of the tree size prior to searching, making the optimization faster.

The selection of the word "terminate" is intended to be contrasted with "eliminate," which is used to describe rotamers that are analogously discarded by using the DEE criterion. Indeed, many of the same rotamers are discarded. As with DEE, termination may be performed iteratively until no further rotamers are terminated. Iterative termination is executed as the preprocessing step before search of the tree.

Although termination serves as an effective preprocessing step, the hallmark of the B&T algorithm is that termination is employed at every level of recursion. At any point of the search, the rotamers defined at levels above the level of the current amino-acid position may be considered a root comprised of a single, partially specified super-rotamer. Termination, then, consists of temporarily considering each of the rotamers at all the remaining positions as candidates for the next appendage of the super-rotamer and applying the bounding criterion to each one. All rotamers terminated this way may be discarded from the optimization of the sub-tree with this partially specified super-rotamer root.

In contrast, the recursive step in a B&B search consists of application of the bounding criterion to the rotamers at only one amino acid position. The benefits of the extra reductions in the sizes of sub-trees far outweigh the costs of calculation of extra bounds for termination. The resulting increase in efficiency makes the B&T search significantly faster than a similarly constructed B&B search.

In a preferred embodiment, it is not necessary to perform iterative termination at every level of recursion, unlike termination preprocessing. A single iteration per branch generally yields the best performance.

In addition, the energetic information produced by the termination process can be used to determine the optimal search order for the remainder of the tree. Because termination effectively replaces the usual bounding process, the resulting breadth-first algorithm is called "Branch-and-Terminate." We also describe a variation of the B&T method that can rapidly find approximate solutions close to the GMEC.

When traversing the combinatorial tree, it is necessary to determine (1) the order in which to explore rotamers at each position, and (2) the sequence in which to explore the different positions. For both cases, we utilize the bounding energies calculated for each rotamer during termination.

We have observed an empirical correlation between low bounding energy and membership in the GMEC. Therefore, the rotamers at each position are searched in order of increasing bounding energy. Conducting the search in this way increases the chance that solutions close to the GMEC are found quickly, thereby providing stringent reference energies early in the calculation.

With respect to the ordering of the different positions, we construct a heuristic based on both the termination bounding energies and the size of the rotamer lists. In a conventional tree search, the positions should be organized in order of increasing number of rotamers per position in order to minimize the total number of nodes in the tree. However, in a B&T search, there are other organization schemes that favor high-level pruning by termination, which reduce the tree size more significantly. We use the bounding energy of the top-ranked (lowest bounding energy) rotamer at each position to indicate which positions are likely to restrict the rest of the system, and consequently favor high-level termination if placed at the super-rotamer root. Because the minimum operators at a node are applied over a set including the subset corresponding to the sub-tree nodes, bounding energies of sub-tree nodes must be higher than or equal to their parent nodes. Therefore, placing positions with high lowest-energies at the top of the tree promotes high bounding energies for their descendents. Since the rotamer lists of a sub-tree can be significantly different than those of its parent, residue ordering is performed at every level of recursion depth.

In a preferred embodiment, an optimal ordering may be obtained by combining energetic and list-size sorting criteria using the following heuristic. Positions are sorted in descending order according to a rank index, as computed in Equation 22,

Equation 22

$$Rank\ Index = (1-f)\frac{1}{1+\ln N} + f\frac{E_{top} - E_{top,min}}{E_{top,max} - E_{top,min}}$$

where N is the number of rotamers at the position, E_{top} is the bounding energy of the top-ranked rotamer of that position, and $E_{top\,min}$ and $E_{top\,max}$ are the minimum and maximum top-ranked bounding energies of all positions, respectively. The expression $1/(1+\ln N)$ is constructed to produce an attenuated weighting inversely proportional to the number of rotamers that evaluates to unity when $N=1$. The quantity f is selected to control the relative weighting of the two criteria. A value of zero for f corresponds to sort based entirely on the number of residues per position, and a value of one produces a ranking based entirely on bounding energies.

A solution that is very close to the GMEC sequence can be found very rapidly by using an approximate variation of the B&T method. Approximate calculations are particularly useful for providing a fast way to obtain low reference energies for exact B&T optimizations. Moreover, the approximate calculation is often sufficient to produce the GMEC energy.

The approximation is based on the observation that the GMEC rotamers are often among those with the lowest termination bounding energies according to the bounding expression. This indicates that the bounding expression has predictive properties. To rapidly find an approximate solution, the ranked rotamer lists are arbitrarily truncated after the pre-processing termination step, and the B&T search is conducted on the abbreviated set of rotamers.

A more reliable solution can be found by repeating the approximate optimization with more lenient truncation, using the solution from the preceding run for the initial reference energy.

The description of the Branch-and-Terminate algorithm herein is tailored for rotamer selection, but the algorithm is in fact generalizable to any combinatorial optimization problem in which all the interactions energies are pairwise and pre-computable. The bounding expression we describe is similarly general.

Although the B&T algorithm can be used by itself, greater benefit can often be obtained by using it in concert with a DEE algorithm. Together, the algorithms can solve optimization problems much more quickly than either can accomplish alone. This may make it possible to quickly find the GMEC for protein design problems that were previously insoluble by either algorithm. Perhaps the most practical use of the B&T algorithm is to complement DEE when dealing with optimization problems that are too difficult to solve using either algorithm alone.

In a preferred embodiment, the B&T and DEE algorithms are used in succession. DEE is used to eliminate rotamers and to perform unification until the optimization reaches iterations that are

inefficient. Inefficiency typically occurs after several unifications when the total number of rotamers and unified super-rotamers gets very large (>5000) and very few eliminations result even from lengthy Goldstein doubles calculations. At this stage, the DEE optimization is aborted, and the state information is transferred to a B&T implementation. Rotamer lists and energy tables are transferred directly, including references to unified super-rotamers, which are transparently represented as ordinary rotamers in the B&T algorithm.

An additional performance improvement is obtained by also passing the list of dead-ending pairs (DEP). DEP's are pairs of rotamers (or super-rotamers) whose members cannot simultaneously exist in the GMEC. These pairs may therefore be safely omitted from the minimum operators in Equation 39.

In a preferred embodiment, the computational processing includes one or more Dead-End Elimination (DEE) computational steps. The DEE theorem is the basis for a very fast discrete search program that was designed to pack protein side chains on a fixed backbone with a known sequence. See Desmet, *et al.*, Nature **356**:539-542 (1992); Desmet, *et al.*, The Proetin Folding Problem and Tertiary Structure Prediction, Ch. **10**:1-49 (1994); Goldstein, Biophys. Jour. **66**:1335-1340 (1994), all of which are incorporated herein by reference. DEE is based on the observation that if a rotamer can be eliminated from consideration at a particular position, i.e. make a determination that a particular rotamer is definitely not part of the global optimal conformation, the size of the search is reduced. This is done by comparing the worst interaction (i.e. energy or E_{total}) of a first rotamer at a single variable position with the best interaction of a second rotamer at the same variable position. If the worst interaction of the first rotamer is still better than the best interaction of the second rotamer, then the second rotamer cannot possibly be in the optimal conformation of the sequence. The original DEE theorem is shown in Equation 23:

Equation 23

$$E(i_a) + \sum_j [\min \text{ over } t\{E(i_a, j_t)\}] > E(i_b) + \sum_j [\max \text{ over } t\{E(i_b, j_t)\}]$$

In Equation 23, rotamer i_a is being compared to rotamer i_b. The left side of the inequality is the best possible interaction energy (E_{total}) of i_a with the rest of the protein; that is, "min over t" means find the rotamer t on position j that has the best interaction with rotamer i_a. Similarly, the right side of the inequality is the worst possible (max) interaction energy of rotamer i_b with the rest of the protein. If this inequality is true, then rotamer i_a is Dead-Ending and can be Eliminated. The speed of DEE comes from the fact that the theorem only requires sums over the sequence length to test and eliminate rotamers.

In a preferred embodiment a variation of DEE is performed. Goldstein DEE, based on Goldstein, (1994) (supra), hereby expressly incorporated by reference, is a variation of the DEE computation, as shown in Equation 24:

Equation 24

$$E(i_a) - E(i_b) + \sum[\text{min over } t\{E(i_a, j_t) - E(i_b, j_t)\}] > 0$$

In essence, the Goldstein Equation 24 says that a first rotamer a of a particular position i (rotamer i_a) will not contribute to a local energy minimum if the energy of conformation with i_a can always be lowered by just changing the rotamer at that position to i_b, keeping the other residues equal. If this inequality is true, then rotamer i_a is Dead-Ending and can be Eliminated.

Thus, in a preferred embodiment, a first DEE computation is done where rotamers at a single variable position are compared, ("singles" DEE) to eliminate rotamers at a single position. This analysis is repeated for every variable position, to eliminate as many single rotamers as possible. In addition, every time a rotamer is eliminated from consideration through DEE, the minimum and maximum calculations of Equation 23, depending on which DEE variation is used, thus conceivably allowing the elimination of further rotamers. Accordingly, the singles DEE computation can be repeated until no more rotamers can be eliminated; that is, when the inequality is not longer true such that all of them could conceivably be found on the global optimum.

In a preferred embodiment, "doubles" DEE is additionally done. In doubles DEE, pairs of rotamers are evaluated; that is, a first rotamer at a first position and a second rotamer at a second position are compared to a third rotamer at the first position and a fourth rotamer at the second position, either using original or Goldstein DEE. Pairs are then flagged as nonallowable, although single rotamers cannot be eliminated, only the pair. Again, as for singles DEE, every time a rotamer pair is flagged as nonallowable, the minimum calculations of Equation 24 change (depending on which DEE variation is used) thus conceivably allowing the flagging of further rotamer pairs. Accordingly, the doubles DEE computation can be repeated until no more rotamer pairs can be flagged.

In addition, in a preferred embodiment, rotamer pairs are initially prescreened to eliminate rotamer pairs prior to DEE. This is done by doing relatively computationally inexpensive calculations to eliminate certain pairs up front. This may be done in several ways, as is outlined below.

To search exhaustively for all dead-ending rotamers at a residue position i, it is necessary to compare every rotamer to every other rotamer available at i. In a comparison matrix, each column corresponds to a particular rotamer, i_r, as a candidate r for elimiantion, and each row corresponds to one of the possible reference rotamers i_t. If there are n t rotamers at position i, then an exhaustive search of n^2-n matrix elements is necessary. Such a matrix is evaluated for each of the positions that may be represented by i.

In a preferred embodiment, the rotamer pair with the lowest interaction energy with the rest of the system is found. Inspection of the energy distributions in sample comparison matrices has revealed that an $i_u j_v$ pair that dead-end eliminates a particular $i_r j_s$ pair can also eliminate other $i_r j_s$ pairs. In fact, there are often a few $i_u j_v$ pairs, which we call "magic bullets," that eliminate a significant number of $i_r j_s$ pairs. We have found that one of the most potent magic bullets is the pair for which maximum interaction energy, $\epsilon_{max}([i_u j_v])k_t$, is least (see Equations 29-31). This pair is referred to as $(i_u j_v)_{mb}$ If this rotamer pair is used in the first round of doubles DEE, it tends to eliminate pairs faster.

Our first speed enhancement is to evaluate the first-order doubles calculation for only the matrix elements in the row corresponding to the $(i_u j_v)_{mb}$ pair. The discovery of $(i_u j_v)_{mb}$ is an n^2 calculation (n = the number of rotamers per position), and the application of Equation 24 to the single row of the matrix corresponding to this rotamer pair is another n^2 calculation, so the calculation time is small in comparison to a full Goldstein calculation. In practice, this calculation produces a large number of dead-ending pairs, often enough to proceed to the next iteration of singles elimination without any further searching of the doubles matrix.

The magic bullet Goldstein calculation will also discover all dead-ending pairs that would be discovered by the Equation 23 or 24, thereby making it unnecessary. This stems from the fact that $\epsilon_{max}((i_u j_v)_{mb})$ must be less than or equal to any $\epsilon_{max}([i_u j_v])$ that would successfully eliminate a pair by Equations 23 or 24.

Since the minima and maxima of any given pair has been precalculated as outlined herein, a second speed-enhancement precalculation may be done. By comparing extrema, pairs that will not dead end can be identified and thus skipped, reducing the time of the DEE calculation. Thus, pairs that satisfy either one of the following criteria are skipped:

Equation 25

$$\epsilon_{min}([i_r j_s]) < \epsilon_{min}([i_u j_v])$$

Equation 26:

$$\epsilon_{max}([i_r j_s]) < \epsilon_{max}([i_u j_v])$$

Because the matrix containing these calculations is symmetrical, half of its elements will satisfy the first inequality Equation 25, and half of those remaining will satisfy the other inequality Equation 26. These three quarters of the matrix need not be subjected to the evaluation of Equation 23 or 24, resulting in a theoretical speed enhancement of a factor of four.

The last DEE speed enhancement refines the search of the remaining quarter of the matrix. This is done by constructing a metric from the precomputed extrema to detect those matrix elements likely to result in a dead-ending pair.

A metric was found through analysis of matrices from different sample optimizations. We searched for combinations of the extrema that predicted the likelihood that a matrix element would produce a dead-ending pair. Interval sizes (see Figure 4) for each pair were computed from differences of the extrema. The size of the overlap of the $i_r j_s$ and $i_u j_v$ intervals were also computed, as well as the difference between the minima and the difference between the maxima. Combinations of these quantities, as well as the lone extrema, were tested for their ability to predict the occurrence of dead-ending pairs. Because some of the maxima were very large, the quantities were also compared logarithmically.

Most of the combinations were able to predict dead-ending matrix elements to varying degrees. The best metrics were the fractional interval overlap with respect to each pair, referred to herein as q_{rs} and q_{uv}.

Equation 27

$$q_{rs} = \frac{interval\ overlap}{interval([i_r j_s])} = \frac{\epsilon_{max}([i_u j_v]) - \epsilon_{min}([i_r j_s])}{\epsilon_{max}([i_r j_s]) - \epsilon_{min}([i_r j_s])}$$

Equation 28

$$q_{uv} = \frac{interval\ overlap}{interval([i_u j_v])} = \frac{\epsilon_{max}([i_u j_v]) - \epsilon_{min}([i_r j_s])}{\epsilon_{max}([i_u j_v]) - \epsilon_{min}([i_u j_v])}$$

These values are calculated using the minima and maxima equations 29, 30, 31 and 32 (see Figure 5):

Equation 29

$$\epsilon_{max}([i_r j_s]) = \epsilon([i_r j_s]) + \sum_{k \neq i \neq j}^{t} \max\epsilon([i_r j_s], k_t)$$

Equation 30

$$\epsilon_{min}([i_r j_s]) = \epsilon([i_r j_s]) + \sum_{k \neq i \neq j}^{t} \min\epsilon([i_r j_s], k_t)$$

Equation 31

$$\epsilon_{max}([i_u j_v]) = \epsilon([i_u j_v]) + \sum_{k \neq i \neq j}^{t} \max\epsilon([i_u j_v], k_t)$$

Equation 32

$$\epsilon_{min}([i_u j_v]) = \epsilon([i_u j_v]) + \sum_{k \neq i \neq j}^{t} \min\epsilon([i_u j_v], k_t)$$

These metrics were selected because they yield ratios of the occurrence of dead-ending matrix elements to the total occurrence of elements that are higher than any of the other metrics tested. For example, there are very few matrix elements (~2%) for which $q_{rs} > 0.98$, yet these elements produce 30-40% of all of the dead-ending pairs.

Accordingly, the first-order doubles criterion is applied only to those doubles for which $q_{rs} > 0.98$ and $q_{uv} > 0.99$. The sample data analyses predict that by using these two metrics, as many as half of the dead-ending elements may be found by evaluating only two to five percent of the reduced matrix.

Generally, as is more fully described below, single and double DEE, using either or both of original DEE and Goldstein DEE, is run iteratively until no further elimination is possible. Usually, convergence is not complete, and further elimination must occur to achieve convergence. This is generally done using "super residue" DEE.

In a preferred embodiment, additional DEE computation is done by the creation of "super residues" or "unification", as is generally described in Desmet , Nature **356**:539-542 (1992); Desmet, *et al.*, The Protein Folding Problem and Tertiary Structure Prediction, Ch. **10**:1-49 (1994); Goldstein, *et al.*, supra. A super residue is a combination of two or more variable residue positions which is then treated as a single residue position. The super residue is then evaluated in singles DEE, and doubles DEE, with either other residue positions or super residues. The disadvantage of super residues is that there are many more rotameric states which must be evaluated; that is, if a first variable residue position has 5 possible rotamers, and a second variable residue position has 4 possible rotamers, there are 20 possible super residue rotamers which must be evaluated. However, these super residues may be eliminated similar to singles, rather than being flagged like pairs.

The selection of which positions to combine into super residues may be done in a variety of ways. In general, random selection of positions for super residues results in inefficient elimination, but it can be done, although this is not preferred. In a preferred embodiment, the first evaluation is the selection of positions for a super residue is the number of rotamers at the position. If the position has too many rotamers, it is never unified into a super residue, as the computation becomes too unwieldy. Thus, only positions with fewer than about 100,000 rotamers are chosen, with less than about 50,000 being preferred and less than about 10,000 being especially preferred.

In a preferred embodiment, the evaluation of whether to form a super residue is done as follows. All possible rotamer pairs are ranked using Equation 33, and the rotamer pair with the highest number is chosen for unification:

Equation 33

$$\frac{\text{fraction of flagged pairs}}{\log(\text{number of super rotamers resulting from the potential unification})}$$

Equation 33 is looking for the pair of positions that has the highest fraction or percentage of flagged pairs but the fewest number of super rotamers. That is, the pair that gives the highest value for Equation 33 is preferably chosen. Thus, if the pair of positions that has the highest number of flagged pairs but also a very large number of super rotamers (that is, the number of rotamers at position i times the number of rotamers at position j), this pair may not be chosen (although it could) over a lower percentage of flagged pairs but fewer super rotamers.

In an alternate preferred embodiment, positions are chosen for super residues that have the highest average energy; that is, for positions i and j, the average energy of all rotamers for i and all

rotamers for j is calculated, and the pair with the highest average energy is chosen as a super residue.

Super residues are made one at a time, preferably. After a super residue is chosen, the singles and doubles DEE computations are repeated where the super residue is treated as if it were a regular residue. As for singles and doubles DEE, the elimination of rotamers in the super residue DEE will alter the minimum energy calculations of DEE. Thus, repeating singles and/or doubles DEE can result in further elimination of rotamers.

Figure 3 is a detailed illustration of the processing operations associated with a ranking module 34 of the invention. The calculation and storage of the singles and doubles energies 70 is the first step, although these may be recalculated every time. Step 72 is the optional application of a cutoff, where singles or doubles energies that are too high are eliminated prior to further processing. Either or both of original singles DEE 74 or Goldstein singles DEE 76 may be done, with the elimination of original singles DEE 74 being generally preferred. Once the singles DEE is run, original doubles (78) and/or Goldstein doubles (80) DEE is run. Super residue DEE is then generally run, either original (82) or Goldstein (84) super residue DEE. This preferably results in convergence at a global optimum sequence. As is depicted in Figure 3, after any step any or all of the previous steps can be rerun, in any order.

The addition of super residue DEE to the computational processing, with repetition of the previous DEE steps, generally results in convergence at the global optimum. Convergence to the global optimum is guaranteed if no cutoff applications are made, although generally a global optimum is achieved even with these steps. In a preferred embodiment, DEE is run until the global optimum sequence is found. That is, the set of optimized protein sequences contains a single member, the global optimum.

In a preferred embodiment, the various DEE steps are run until a managable number of sequences is found, i.e. no further processing is required. These sequences represent a set of optimized protein sequences, and they can be evaluated as is more fully described below. Generally, for computational purposes, a manageable number of sequences depends on the length of the sequence, but generally ranges from about 1 to about 10^{15} possible rotamer sequences. This range can be extended to approximately 10^{30} if B&T is used as the next analyzing step.

Alternatively, DEE is run to a point, resulting in a set of optimized sequences (in this context, a set of remainder sequences) and then further compututational processing of a different type may be

run. For example, in one embodiment, direct calculation of sequence energy as outlined above is done on the remainder possible sequences. Alternatively, a Monte Carlo search can be run. In another embodiment, B&T can be run.

In a preferred embodiment, the computation processing need not comprise a DEE computational step. In this embodiment, a Monte Carlo search is undertaken, as is known in the art. See Metropolis *et al.*, J. Chem. Phys. 21:1087 (1953), hereby incorporated by reference. In this embodiment, a random sequence comprising random rotamers is chosen as a start point. In one embodiment, the variable residue positions are classified as core, boundary or surface residues and the set of available residues at each position is thus defined. Then a random sequence is generated, and a random rotamer for each amino acid is chosen. This serves as the starting sequence of the Monte Carlo search. A Monte Carlo search then makes a random jump at one position, either to a different rotamer of the same amino acid or a rotamer of a different amino acid, and then a new sequence energy ($E_{\text{total sequence}}$) is calculated, and if the new sequence energy meets the Boltzmann criteria for acceptance, it is used as the starting point for another jump. If the Boltzmann test fails, another random jump is attempted from the previous sequence. In this way, sequences with lower and lower energies are found, to generate a set of low energy sequences.

If computational processing results in a single global optimum sequence, it is frequently preferred to generate additional sequences in the energy neighborhood of the global solution, which may be ranked. These additional sequences are also optimized protein sequences. The generation of additional optimized sequences is generally preferred so as to evaluate the differences between the theoretical and actual energies of a sequence. Generally, in a preferred embodiment, the set of sequences is at least about 75% homologous to each other, with at least about 80% homologous being preferred, at least about 85% homologous being particularly preferred, and at least about 90% being especially preferred. In some cases, homology as high as 95% to 98% is desirable. Homology in this context means sequence similarity or identity, with identity being preferred. Identical in this context means identical amino acids at corresponding positions in the two sequences which are being compared. Homology in this context includes amino acids which are identical and those which are similar (functionally equivalent). This homology will be determined using standard techniques known in the art, such as the Best Fit sequence program described by Devereux, *et al.*, Nucl. Acid Res., **12**:387-395 (1984), or the BLASTX program (Altschul, *et al.*, J. Mol. Biol., **215**:403-410 (1990)) preferably using the default settings for either. The alignment may include the introduction of gaps in the sequences to be aligned. In addition, for sequences which contain either more or fewer amino acids than an optimum sequence, it is understood that the percentage of homology will be determined based on the number of homologous amino acids in relation to the total number of amino acids. Thus, for example, homology of sequences shorter than an optimum will be determined using the number of amino acids in the shorter sequence.

Once optimized protein sequences are identified, the processing of Figure 2 optionally proceeds to step 56 which entails searching the protein sequences. This processing may be implemented with the search module 36. The search module 36 is a set of computer code that executes a search strategy. For example, the search module 36 may be written to execute a Monte Carlo search as described above. Starting with the global solution, random positions are changed to other rotamers allowed at the particular position, both rotamers from the same amino acid and rotamers from different amino acids. A new sequence energy ($E_{total\ sequence}$) is calculated, and if the new sequence energy meets the Boltzmann criteria for acceptance, it is used as the starting point for another jump. See Metropolis *et al.*, 1953, supra, hereby incorporated by reference. If the Boltzmann test fails, another random jump is attempted from the previous sequence. A list of the sequences and their energies is maintained during the search. After a predetermined number of jumps, the best scoring sequences may be output as a rank-ordered list. Preferably, at least about 10^6 jumps are made, with at least about 10^7 jumps being preferred and at least about 10^8 jumps being particularly preferred Preferably, at least about 100 to 1000 sequences are saved, with at least about 10,000 sequences being preferred and at least about 100,000 to 1,000,000 sequences being especially preferred. During the search, the temperature is preferably set to 1000 K.

Once the Monte Carlo search is over, all of the saved sequences are quenched by changing the temperature to 0 K, and fixing the amino acid identity at each position. Preferably, every possible rotamer jump for that particular amino acid at every position is then tried.

The computational processing results in a set of optimized protein sequences. These optimized protein sequences are generally, but not always, significantly different from the wild-type sequence from which the backbone was taken. That is, each optimized protein sequence preferably comprises at least about 5-10% variant amino acids from the starting or wild-type sequence, with at least about 15-20% changes being preferred and at least about 30% changes being particularly preferred

These sequences can be used in a number of ways. In a preferred embodiment, one, some or all of the optimized protein sequences are constructed into designed proteins, as show with step 58 of Figure 2. Thereafter, the protein sequences can be tested, as shown with step 60 of the Figure 2. Generally, this can be done in one of two ways.

In a preferred embodiment, the designed proteins are chemically synthesized as is known in the art. This is particularly useful when the designed proteins are short, preferably less than 150 amino acids in length, with less than 100 amino acids being preferred, and less than 50 amino acids being

particularly preferred, although as is known in the art, longer proteins can be made chemically or enzymatically.

In a preferred embodiment, particularly for longer proteins or proteins for which large samples are desired, the optimized sequence is used to create a nucleic acid such as DNA which encodes the optimized sequence and which can then be cloned into a host cell and expressed. Thus, nucleic acids, and particularly DNA, can be made which encodes each optimized protein sequence. This is done using well known procedures. The choice of codons, suitable expression vectors and suitable host cells will vary depending on a number of factors, and can be easily optimized as needed.

Once made, the designed proteins are experimentally evaluated and tested for structure, function and stability, as required. This will be done as is known in the art, and will depend in part on the original protein from which the protein backbone structure was taken. Preferably, the designed proteins are more stable than the known protein that was used as the starting point, although in some cases, if some constaints are placed on the methods, the designed protein may be less stable. Thus, for example, it is possible to fix certain residues for altered biological activity and find the most stable sequence, but it may still be less stable than the wild type protein. Stable in this context means that the new protein retains either biological activity or conformation past the point at which the parent molecule did. Stability includes, but is not limited to, thermal stability, i.e. an increase in the temperature at which reversible or irreversible denaturing starts to occur; proteolytic stability, i.e. a decrease in the amount of protein which is irreversibly cleaved in the presence of a particular protease (including autolysis); stability to alterations in pH or oxidative conditions; chelator stability; stability to metal ions; stability to solvents such as organic solvents, surfactants, formulation chemicals; etc.

In a preferred embodiment, the modelled proteins are at least about 5% more stable than the original protein, with at least about 10% being preferred and at least about 20-50% being especially preferred.

The results of the testing operations may be computationally assessed, as shown with step 62 of Figure 2. An assessment module 38 may be used in this operation. That is, computer code may be prepared to analyze the test data with respect to any number of metrices.

At this processing juncture, if the protein is selected (the yes branch at block 64) then the protein is utilized (step 66), as discussed below. If a protein is not selected, the accumulated information

may be used to alter the ranking module 34, and/or step 56 is repeated and more sequences are searched.

In a preferred embodiment, the experimental results are used for design feedback and design optimization.

Once made, the proteins of the invention find use in a wide variety of applications, as will be appreciated by those in the art, ranging from industrial to pharmocological uses, depending on the protein. Thus, for example, proteins and enzymes exhibiting increased thermal stability may be used in industrial processes that are frequently run at elevated temperatures, for example carbohydrate processing (including saccharification and liquifaction of starch to produce high fructose corn syrup and other sweetners), protein processing (for example the use of proteases in laundry detergents, food processing, feed stock processing, baking, etc.), etc Similarly, the methods of the present invention allow the generation of useful pharmaceutical proteins, such as analogs of known proteinaceous drugs which are more thermostable, less proteolytically sensitive, or contain other desirable changes.

The following examples serve to more fully describe the manner of using the above-described invention, as well as to set forth the best modes contemplated for carrying out various aspects of the invention. It is understood that these examples in no way serve to limit the true scope of this invention, but rather are presented for illustrative purposes. All references cited herein are explicitly incorporated by reference in their entirety.

EXAMPLES

Example 1

The Generation and Use of B&T

We tested the generality of the algorithm by applying it to a suite of optimization problems representative of different protein structural classes. Rotamers were selected from a backbone dependent library (Dunbrack, R.L. & Karplus, M., *J. Mol. Biol.* **230**, 543-574 (1993)). To test α-helical surface positions, the 12 residues occupying the **b**, **c**, and **f** locations in the heptad repeat of one helix of the coiled-cold GCN4-p1 dimer (E.K. O'Shea, et al., *Science.* **254**, 539-544 (1991)) were optimized from the set of rotamers corresponding to hydrophilic amino acids (A, D, E, H, K, N, Q, R, S, and T). There were 9.1×10^{22} rotameric combinations.

The β1 domain of streptococcal protein G (Gallagher, T. et al., *Biochemistry*. 33, 4721-4728 (1994)) was used for the remaining cases. As a representative of core and boundary optimization problems, a subset of positions determined to be in the core and boundary according to our residue classification scheme (positions 3, 5, 7, 12, 23, 25, 26, 30, 34, 43, 45, 52, 54) were optimized from the 3.4×10^{25} combinations of hydrophobic rotamers (amino acids A, F, I, L, M, V, W, and Y). For β-sheet surfaces, a subset of the β-sheet surface residues (positions 4, 6, 15, 17, 42, 44, 53, 55) were optimized from the 4.9×10^{17} combinations of hydrophilic rotamers.

To represent problems consisting of a mixture of different structural types, including turns, we also included the optimization of the residues containing any atoms within 10 Å of the side-chain atoms of Val 21. Of these fourteen, the core residues (positions 3, 20, 36) were allowed to have any of the hydrophobic identities, the surface residues (positions 2, 19, 21, 22, 24) had hydrophilic identities, and the remaining boundary residues (positions 1, 18, 23, 25, 27, 29) were selected from a group of hydrophilic and hydrophobic residues, excluding methionine (amino acids A, D, E, F, H, I, K, L, N, Q, R, S, T, V, W, and Y). There were 1.3×10^{29} possible rotameric combinations.

The most difficult benchmark consisted of all 18 non-glycine core and boundary residues (Malakaukas S. & Mayo, S.L., *Nat. Struc. Biol.* 5, 470-475 (1998)). The core residues (positions 3, 5, 7, 20, 26, 30, 34, 39, 52, 54) were selected from the set of hydrophobic amino acids, and the boundary residues (positions 1, 12, 23, 33, 37, 45, 50, 56) were selected from the composite list of hydrophilic and hydrophobic residues. There were 1.9×10^{34} possible rotameric combinations.

Energy Expression

We employ an energy expression that consists of van der Waals, electrostatic, and solvation terms. For van der Waals, a Lennard-Jones 6-12 potential was used, with radii scaled by a factor of 0.9 (Dahiyat B.I. & Mayo, S.L., *Proc. Natl. Acad. Sci. USA*. **94**, 10172-10177 (1997)). Electrostatics were computed using a distance dependent dielectric and a hybridization-dependent hydrogen-bonding term (Dahiyat, B.I., et al., *Protein Science*. **6**, 1333-1337 (1997)). Solvation effects were approximated from hydrophobic surface area burial (Street, A.G. & Mayo, S.L., *Folding & Design*. **3**, 253-258 (1998)). Atom radii and hydrogen-bond well depths were based on the DREIDING force-field (Mayo, S.L., et al., *J. Phys. Chem.* **94**, 8897-8909 (1990)).

Calculation

For reference, calculation times were recorded using a fully optimized DEE algorithm incorporating high energy threshold reduction (HETR) (De Maeyer, M., et al., *Folding & Design*, **2**, 53-56 (1997)) and magic bullets and other doubles optimizations (Gordon, D.B. & Mayo, S.L., *J. Comp. Chem.* **19**,

1505-1514. (1998)). Calculations were also performed using an enhanced B&B implementation that employed the efficient bounding criteria and termination preprocessing.

For the first three benchmark cases, all calculations were performed using an initial upper bound of 0.0 kcal/mol, since our energy expression typically results in optimal sequences with negative energies. For the remaining two cases, initial bounds were obtained by first running the approximate version of the algorithm, in which the rotamer lists were truncated to the fifteen rotamers with the lowest bounding energies at each residue position. These provided initial bounds of –153.0 and –250.0 kcal/mol, respectively.

The generality of the sorting criteria was demonstrated by performing optimizations with values of *f* in Equation 5 ranging from 0 to 1.

To illustrate the reliability of the approximate form of the algorithm, optimizations were also performed using only the top 30 rotamers at each position as ranked after a single round of termination.

The larger benchmark problem consisting of core and boundary residues was used to demonstrate how DEE and B&T methods can work in concert. The problem was optimized using a DEE algorithm, and upon every reduction of complexity by at least and order of magnitude, the state of the diminished problem was recorded. A B&T algorithm was used to complete the calculation for each reduced state. The calculations were performed using the optimal sorting factor as determined from the previous benchmarks.

For all calculations, the total CPU time was recorded as well as the portions of that time spent performing termination preprocessing and the actual recursive search. The total number of nodes comprising the final pruned tree was also recorded by tallying the number of nodes remaining after termination at every level of recursion. Calculations were performed on a single R10000 CPU of a Silicon Graphics Origin 2000.

Pairwise Bounding Expression

This section describes the construction of a stringent expression for a lower bound for a system composed only of one and two-body interactions in terms of both a partially specified sequence and the set of rotamers available at its unspecified positions.

For a system consisting only of two-body interactions, the total potential energy can be expressed as the sum of energies between all pairs

Equation 34

$$E_{\text{total}} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E(i,j) = \sum_{i} \sum_{\substack{j \\ j>i}} E(i,j)$$

In a protein, *i* and *j* refer to amino-acid positions, and *E(i,j)* is energy of interaction between amino-acids at those positions.

A protein system also consists of single-body interactions. Because each body is an amino-acid side chain at a particular position on the protein backbone, there is an energy contribution both from side chain interactions with other side chains as well as interactions with the protein template scaffolding. Both energies of interaction depend on the side chain position, amino acid identity, and configuration. Thus the total potential energy can be expressed,

Equation 35

$$E_{\text{total}} = \sum_{i} E(i_c, \text{template}) + \sum_{i} \sum_{\substack{j \\ j>i}} E(i_c, j_c)$$

where, and *c* is a position-specific index describing a side chain rotamer of a particular amino acid type and configuration.

For the purposes of deriving an expression for a lower bound, it is desirable to alter the indices to allow redundancy.

Equation 36

$$E_{\text{total}} = \sum_{i} E(i_c, \text{template}) + \tfrac{1}{2} \sum_{i} \sum_{\substack{j \\ j \neq i}} E(i_c, j_c)$$

To ensure that the bounding expression satisfies the condition in Equation 3, we use the following inequalities (Equations 37 and 38):

Equation 37

$$\min_{r} \left[E(i_r, \text{template}) \right] \leq E(i_g, \text{template})$$

Equation 38

$$\min_{r} \left[E(i_r, j_g) \right] \leq E(i_g, j_g)$$

in which the indices *r* and *s* refer to all of the possible rotamers available at each position, and the minimum operator selects the single rotamer that minimizes the sub-expression. The index *g* denotes the rotamer found at the specified position in the global minimum combination. A simple expression for the lower bound is therefore obtained by summing minimal interaction energies between positions by discovering minimal rotamer-pairs.

Equation 39

$$E^{(0)}_{\text{bound}} = \sum_i \min_r \left[E(i_r, \text{template})\right] + \frac{1}{2}\sum_i \min_r \sum_{j \neq i} \min_s \left[E(i_r, j_s)\right]$$

The derivation above represents a generic strategy for producing a bounding expression from any total energy expression. For example, more restrictive bounds can be obtained from energy expressions that sum over three or four-body interactions. However, the computational cost to implement such bounds on a protein system is very high. Fortunately, there are variations of Equation 35 that are equivalent in terms of computational cost yet yield better bounds.

An alternative way to express the total energy of the system is to distribute the template energies into the pair calculation. Given an energy quantity for a pair of rotamers,

Equation 40

$$E_{\text{pair}}(i_c, j_c) = \frac{E(i_c, \text{template}) + E(j_c, \text{template})}{2p - 2} + \frac{E(i_c, j_c)}{2}$$

in which *p* is the number of amino acid positions, the total energy can be expressed,

Equation 41

$$E_{\text{total}} = \sum_i \sum_{j \neq i} E_{\text{pair}}(i_c, j_c)$$

which, in turn, can be used to produce the following bounding expression as shown in Equation 42,

Equation 42

$$E^{(1)}{}_{bound} = \sum_i \min_r \sum_{j, j \neq 1} \min_s [E_{pair}(i_r, j_s)]$$

Because the minima must be evaluated with respect to single-body and pair energies simultaneously, this bounding expression is necessarily greater than or equal to the expression in Equation 11. Therefore the new bound is more restrictive. The computational requirements for both expressions, however, are of the same order. Each requires n^2p^2 calculations, where *n* is the average number of available rotamers per position, and *p* is the number of positions.

One can derive a lower bound that is even more restrictive by performing an expansion of Equation 41 before applying the minimization operators. When testing a particular node during traversal of the combinatorial tree, the positions corresponding to nodes above (and including) the current node have uniquely specified rotamers, whereas the remaining, deeper nodes are not yet uniquely specified. The set of all amino acid positions can therefore be decomposed into two subsets, fixed (*F*) and variable (*V*). Equation 41 can be rewritten as Equation 43,

Equation 43

$$E_{\text{total}} = \sum_{i \in \{F,V\}} \sum_{\substack{j \in \{F,V\} \\ j \neq i}} E_{\text{pair}}(i_c, j_c)$$

Next, we expand the summation to give Equation 44:

Equation 44

$$E_{\text{total}} = \sum_{i \in F} \sum_{\substack{j \in F \\ j \neq i}} E_{\text{pair}}(i_c, j_c) + \sum_{i \in F} \sum_{j \in V} E_{\text{pair}}(i_c, j_c) + \sum_{i \in V} \sum_{j \in F} E_{\text{pair}}(i_c, j_c) + \sum_{i \in V} \sum_{\substack{j \in V \\ j \neq i}} E_{\text{pair}}(i_c, j_c)$$

Application of the minimum operators to this expression would yield a bounding expression equivalent to Equation 42. To increase the stringency, we utilize the inequality,

Equation 45

$$\min_r \sum_j E_{pair}(i_r, j_s) \geq \sum_j \min_r E_{pair}(i_r, j_s)$$

The middle two terms of Equation 44 differ only in their indices, and are therefore equivalent to one another. However, there is a difference once the minimum operators are applied, since the rotamers of the fixed subset (*F*) will restrict the selection of the minimum energy rotamer pair for the minimized third term, but not for the second. Therefore, we reverse the order of the summation for the second term and combine it with the third term to make use of (Equation 45) such that the minimum will be as large as possible,

Equation 46

$$E_{\text{total}} = \sum_{i \in F} \sum_{\substack{j \in F \\ j \neq i}} E_{\text{pair}}(i_c, j_c) + 2\sum_{i \in V} \sum_{j \in F} E_{\text{pair}}(i_c, j_c) + \sum_{i \in V} \sum_{\substack{j \in V \\ j \neq i}} E_{\text{pair}}(i_c, j_c)$$

Now we apply the minimum operators to all sums over positions that are not uniquely specified.

Equation 47

$$E_{\text{bound}}^{(2)} = \sum_{i \in F} \sum_{\substack{j \in F \\ j \neq i}} E_{\text{pair}}(i_r, j_s) + 2\sum_{i \in V} \min_r \sum_{j \in F} E_{\text{pair}}(i_r, j_s) + \sum_{i \in V} \min_r \sum_{\substack{j \in V \\ j \neq i}} \min_s E_{\text{pair}}(i_r, j_s)$$

To achieve further stringency, we rearrange Equation 23 before applying the minimum operators.

Equation 48

$$E_{\text{total}} = \sum_{i \in F} \sum_{\substack{j \in F \\ j \neq i}} E_{\text{pair}}(i_c, j_c) + \sum_{i \in V} \left\{ 2\sum_{j \in F} E_{\text{pair}}(i_c, j_c) + \sum_{\substack{j \in V \\ j \neq i}} E_{\text{pair}}(i_c, j_c) \right\}$$

From which we obtain, Equation 49

Equation 49

$$E_{\text{bound}}^{(\text{final})} = \sum_{i \in F} \sum_{\substack{j \in F \\ j \neq i}} E_{\text{pair}}(i_r, j_s) + \sum_{i \in V} \min_r \left\{ 2\sum_{j \in F} E_{\text{pair}}(i_r, j_s) + \sum_{\substack{j \in V \\ j \neq i}} \min_s \left[E_{\text{pair}}(i_r, j_s) \right] \right\}$$

The expression is generalizable to any system consisting only of two-body interactions such that the total energy of the system can be expressed as in Equation 41.

Efficient Implementation of Bounding Expression

The computational cost of evaluating Equation 49 is proportional to p^2n^2, where p is the number of positions and n is the average number of rotamers at each position. When performing termination, the bound is evaluated for all pn rotamers, so that the total calculation order for a round of termination is p^3n^3.

Termination consists of evaluating the bounding expression for rotamers at all the unspecified positions. Therefore, a position is temporarily considered a member of set F while its rotamers are being evaluated. Since the expensive second term of the final summation is dependent only on V, its possible values may be precomputed for all rotamers i_r once per position and placed into a table for lookup during the evaluation of Equation 49.

The cost of performing p^2n^2 calculations to assembling the table for the termination of all n positions scales as p^3n^2. The bounding expression now only requires order pn calculations for each of the pn times it is performed, for an overall order of p^2n^2. The overall calculation time therefore scales approximately as p^3n^2, which is nearly n times faster than the direct implementation. Since n is often as large as 100-200, the speed increase can be drastic.

Results

To assess the generality of the B&T approach, different incarnations of the algorithm were applied to benchmark problems representing different structural classes, as described in Materials and methods. Optimization times were heavily dependent on the sorting heuristic, as shown in Figure 8. The performance improvement, as measured by dividing the total optimization times, ranged from a factor of three for the β-sheet case to a factor of over forty for the "mixed" case. Remarkably, very similar values of the sorting factor f produced the fastest optimization times for all structural classes. Initially, values at intervals of 0.1 were tested, but since all benchmark cases exhibited minima near $f = 0.1$, values at intervals of 0.01 were sampled near this value. At this level of refinement, the different cases had different optimal sorting factor values, but a value of $f = 0.08$ was close to optimal for all of them. We also observe that optimizations with the fastest times had the fewest nodes in their pruned combinatorial trees.

The total calculation times for the benchmarks using a sorting factor of 0.08 are competitive compared to times of a highly optimized DEE algorithm, and are significantly faster than the optimized B&B search (Figure 9). For the β-sheet surface and the small core-boundary calculations, the B&T method is approximately twenty times faster than DEE. For the mixed case, it is nearly eight times faster. For the α-helical case, however, the B&T method is more than two times slower. This is likely a reflection of the linear topological arrangement of the system, in which it difficult to select positions to place at the tree root that both restrict large parts of the system and are themselves restricted.

The approximate form of the algorithm proved to be exceptionally effective. For the four cases above, B&T calculations that used only the 30 top-ranked rotamers at each position all took less than fifteen seconds and produced the correct GMEC solutions. For the more difficult core-boundary case, the calculation took five minutes, and also produced the correct GMEC solution. For this case, a more aggressive calculation using only the top 15 rotamers at each position took 25 seconds and produced a solution whose energy was in error by less than 1%. This energy was used as the initial bound for the remaining calculations on the system.

To illustrate the potential for combining DEE and B&T methods by way of DEE preprocessing, we selected a problem computable by either algorithm to enable us to perform quantitative comparisons. In practice, however, the technique is applied to problems that are not currently computable in reasonable computer time by either algorithm, for which the benefit is obviously much greater. Figure 10 illustrates the total calculation times partitioned into DEE and B&T times for optimization of the difficult benchmark consisting of core and boundary residues. The calculations differ in the amount of time allotted to DEE reduction before completion with the B&T algorithm. At the best timing, the combined algorithms complete the optimization eight times faster than DEE alone. Moreover, we have observed that, in practice, the B&T method is generally effective on completing large problems that DEE can reduce to as high as 10^{30} remaining sequences.

Discussion

We have described a deterministic search method for rotamer optimization, and have demonstrated that it is as fast as the current standard algorithm for rotamer optimization in protein design, and in some cases, it is much faster. The success of the Branch-and-Terminate method rests on the construction of a novel pairwise bounding expression, which is used both to perform termination and to supply energetic information with which to determine the search order. Although the algorithm is tailored to protein systems, it is generalizable to any problem that can be similarly described.

Although the B&T algorithm is quite effective when used alone, it is perhaps more important that it increases the problem size limit of DEE calculations by providing an efficient way to complete optimizations for which elimination criteria have become less effective at removing rotamers. This makes it possible to perform optimizations on larger proteins and on systems with large numbers of interacting residues.

The size limit may be raised even higher once the limitations of the approximate form of the algorithm become better understood. For the benchmark cases, the approximate algorithm found the GMEC solutions up to a thousand times faster than either of the exact methods. Even the DEE implementation to which the B&T method is compared incorporates some conservative approximations in the form of high energy threshold rejection (HETR) criteria (De Maeyer, M., et al., *Folding & Design*, **2**, 53-56 (1997)). Analogous techniques may provide a way to construct a faster, approximate B&T algorithm with a clearly defined accuracy. Along the same line of reasoning, truncation based on bounding energies might be an effective replacement for HETR cutoffs in DEE.

There is also room for improvement in the heuristic for determining search order. Heuristics that are even more effective may exist that make use of structural information, in addition to energetics and size considerations.

In addition, we are currently exploring features of the B&T algorithm that are common to all backtrack searches. First, it is possible to exhaustively sample the amino acid and rotamer sequence space near the GMEC. This is accomplished by modifying the algorithm so that it refrains from lowering the initial minimum energy upon finding low energy combinations (Leach, A.R. & Lemon, A.P., *Proteins.* **33**, 227-239 (1998)). The result is a full enumeration of all sequences with energies below the specified initial minimum energy, provided that this energy is close enough to the GMEC energy that the calculation remains tractable

Also, it is straightforward to parallelize backtrack algorithms by dispatching branches to different CPU's. We observe a scaling efficiency between 60%-80%, depending on the type of problem. Another advantage of the tree representation is that it makes it possible to estimate how much time the optimization will require. This is accomplished using a well-known tree estimation technique (Hall, M. & Knuth, D.E., *Amer. Math. Monthly.* **72**, 21-28 (1995)) in which statistics are compiled for randomsample trajectories through the tree. This has helped us to predict when it is best to transfer DEE problems to B&T for completion.

In practice, we believe that the best way to use the B&T method is to first attempt to optimize a problem using DEE. Upon observing that DEE begins to produce very few eliminations or dead-ending pairs, the state information should be transferred to an approximate form of the B&T algorithm. Using the energy from this calculation as the initial upper bound, the approximate algorithm may be repeated again with successively more conservative truncations. The final energy should then be used as the initial bound for the exact B&T calculation.

CLAIMS

We claim:

1. A method executed by a computer under the control of a program, said computer including a memory for storing said program, said method comprising the steps of:

(A) receiving a protein backbone structure with variable residue positions;

(B) establishing a group of potential rotamers for each of said variable residue positions, wherein at least one variable residue position has rotamers from at least two different amino acid side chains; and

(C) analyzing the interaction of each of said rotamers with all or part of the remainder of said protein backbone structure to generate a set of optimized protein sequences, wherein said analyzing step includes a Branched and Terminate (B&T) computation.

2. A method according to claim 1 wherein said analyzing step further comprises a DEE computation.

3. A method according to claim 1 or 2 wherein said set of optimized protein sequences comprises the globally optimal protein sequence.

4. A method according to claim 2 wherein said DEE computation is selected from the group consisting of original DEE and Goldstein DEE.

5. A method according to claim 1 or 2 wherein said analyzing step includes the use of at least one scoring function.

6. A method according to claim 5 wherein said scoring function is selected from the group consisting of a Van der Waals potential scoring function, a hydrogen bond potential scoring function, an atomic solvation scoring function, an electrostatic scoring function and a secondary structure propensity scoring function.

7. A method according to claim 6 wherein said analyzing step includes the use of at least two scoring functions.

8. A method according to claim 6 wherein said analyzing step includes the use of at least three scoring functions.

9. A method according to claim 6 wherein said analyzing step includes the use of at least four scoring functions.

10. A method according to claim 6 wherein said atomic solvation scoring function includes a scaling factor that compensates for over-counting.

11. A method according to claim 1or 2 further comprising testing at least one member of said set to produce experimental results

12. A method according to claim 3 further comprising

(D) generating a rank ordered list of additional optimal sequences from said globally optimal protein sequence.

13. A method according to claim 12 wherein said generating includes the use of a Monte Carlo search.

14. A method according to claim 1 wherein said analyzing step step comprises a Monte Carlo computation.

15. A method according to any of claims 1-14 further comprising:

(E) testing some or all of said protein sequences from said ordered list to produce potential energy test results.

16. A method according to claim 15 further comprising:

(F) analyzing the correspondence between said potential energy test results and theoretical potential energy data.

17. A method according to claim 1 further comprising altering at least one supersecondary structure parameter value of said protein backbone structure prior to establishing said potential rotamer group.

18. An optimized protein sequence generated by the method of claim 1.

19. A nucleic acid sequence encoding a protein sequence according to claim 18.

20. An expression vector comprising the nucleic acid of claim 19.

21. A host cell comprising the nucleic acid of claim 19.

22. A computer readable memory to direct a computer to function in a specified manner, comprising:

a side chain module to correlate a group of potential rotamers for residue positions of a protein backbone model;

a ranking module to analyze the interaction of each of said rotamers with all or part of the remainder of said protein to generate a set of optimized protein sequences wherein said analysis includes a B&T computation.

23. A computer readable memory according to claim 22 wherein said ranking module includes a van der Waals scoring function component.

24. A computer readable memory according to claim 22 wherein said ranking module includes an atomic solvation scoring function component.

25. A computer readable memory according to claim 22 wherein said ranking module includes a hydrogen bond scoring function component.

26. A computer readable memory according to claim 22 wherein said ranking module includes a secondary structure scoring function component.

27. A computer readable memory according to claim 22 further comprising

an assessment module to assess the correspondence between potential energy test results and theoretical potential energy data.

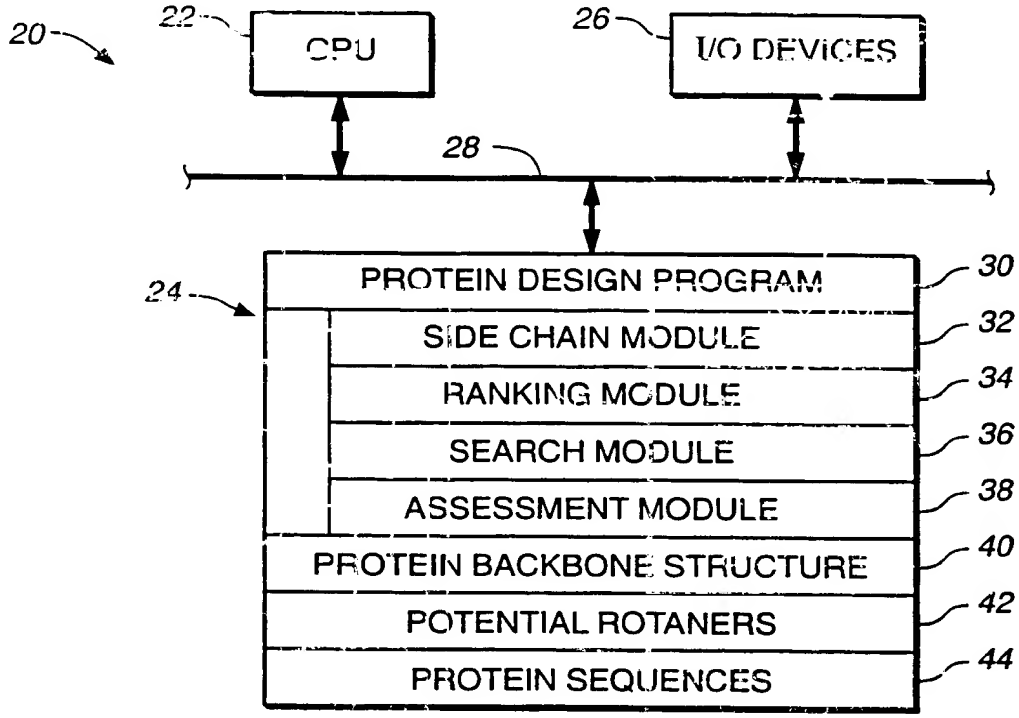

20
22
CPU
26
I/O DEVICES
28
24
PROTEIN DESIGN PROGRAM
30
SIDE CHAIN MODULE
32
RANKING MODULE
34
SEARCH MODULE
36
ASSESSMENT MODULE
38
PROTEIN BACKBONE STRUCTURE
40
POTENTIAL ROTANERS
42
PROTEIN SEQUENCES
44

FIG._1

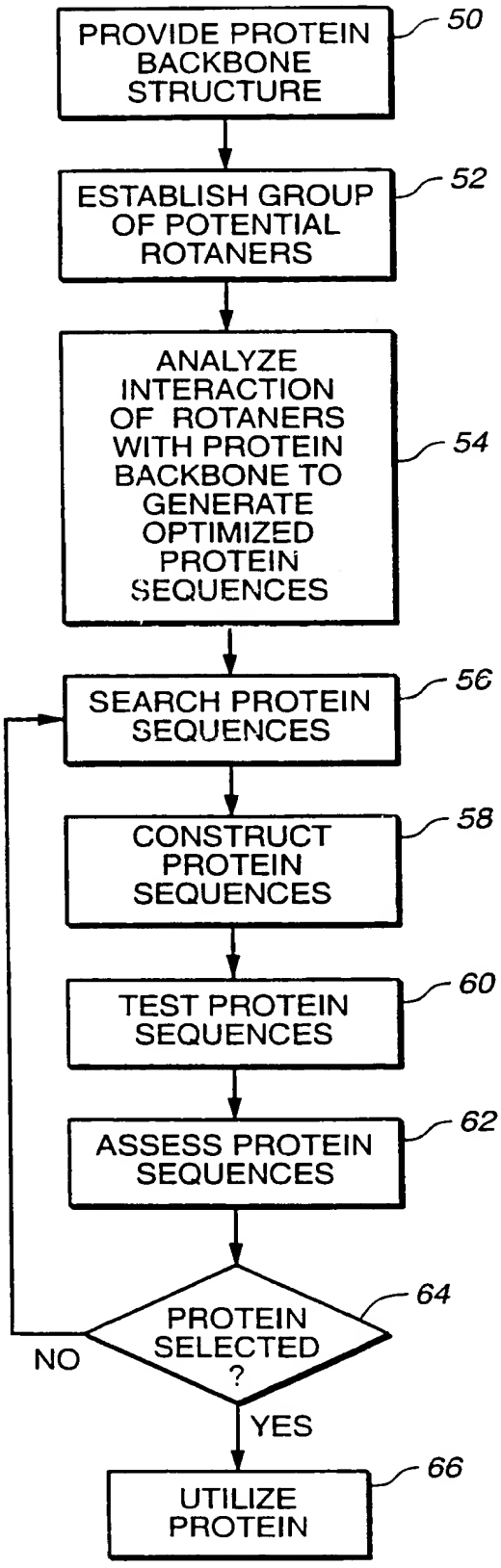

PROVIDE PROTEIN BACKBONE STRUCTURE
50
ESTABLISH GROUP OF POTENTIAL ROTANERS
52
ANALYZE INTERACTION OF ROTANERS WITH PROTEIN BACKBONE TO GENERATE OPTIMIZED PROTEIN SEQUENCES
54
SEARCH PROTEIN SEQUENCES
56
CONSTRUCT PROTEIN SEQUENCES
58
TEST PROTEIN SEQUENCES
60
ASSESS PROTEIN SEQUENCES
62
PROTEIN SELECTED ?
64
NO
YES
UTILIZE PROTEIN
66

FIG._2

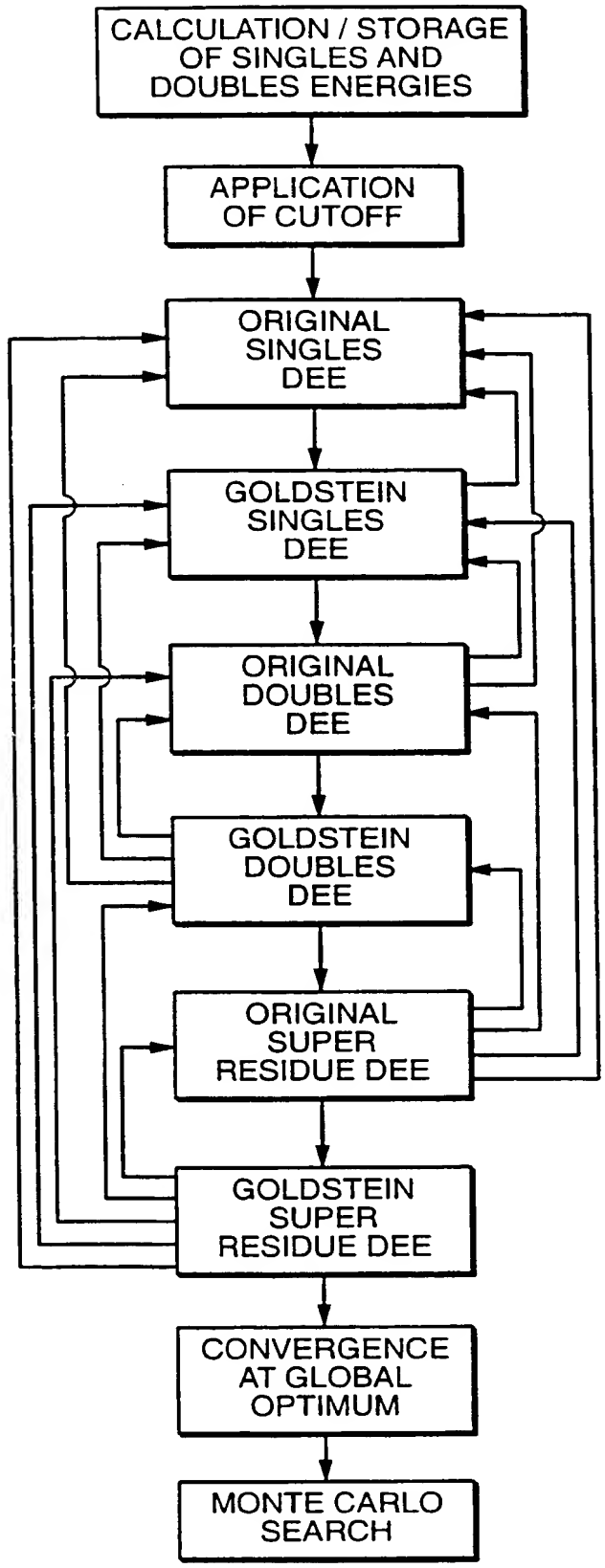

CALCULATION / STORAGE OF SINGLES AND DOUBLES ENERGIES
APPLICATION OF CUTOFF
ORIGINAL SINGLES DEE
GOLDSTEIN SINGLES DEE
ORIGINAL DOUBLES DEE
GOLDSTEIN DOUBLES DEE
ORIGINAL SUPER RESIDUE DEE
GOLDSTEIN SUPER RESIDUE DEE
CONVERGENCE AT GLOBAL OPTIMUM
MONTE CARLO SEARCH

FIG._3

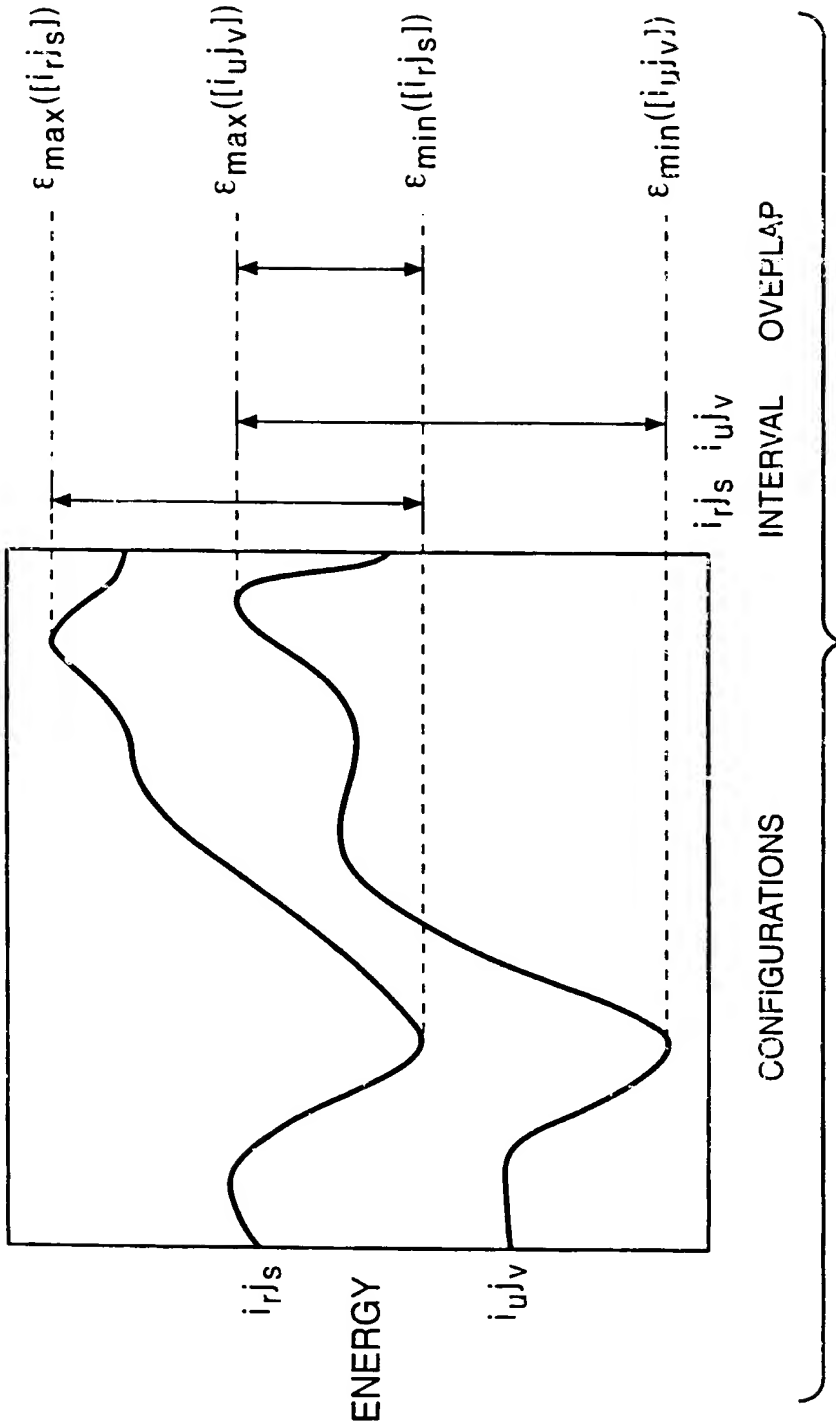

$\varepsilon_{max}([i_r i_s])$
$\varepsilon_{max}([i_u j_v])$
$\varepsilon_{min}([i_r j_s])$
$\varepsilon_{min}([i_u j_v])$
$i_r j_s$ $i_u j_v$
INTERVAL OVERLAP
CONFIGURATIONS
ENERGY
$i_r j_s$
$i_u j_v$

FIG._4

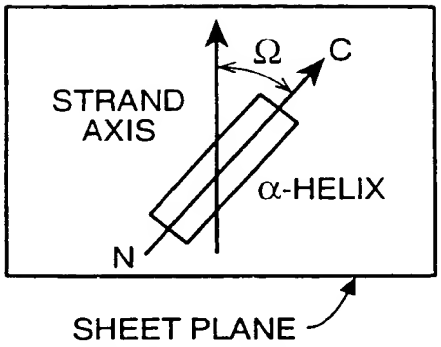


FIG._5A

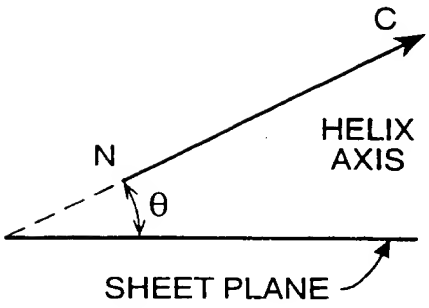


FIG._5B

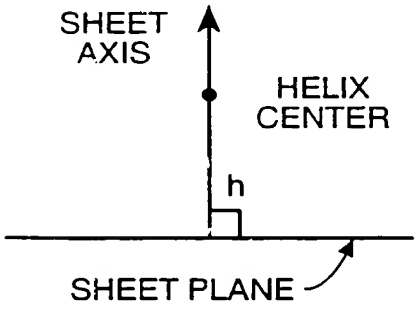


FIG._5C

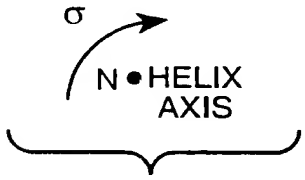


FIG._5D

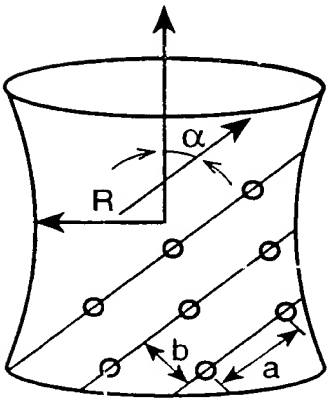


FIG._6A

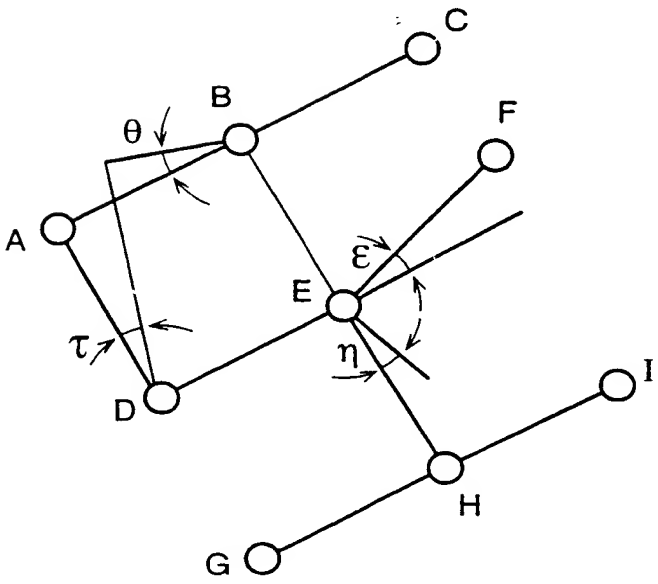


FIG._6B

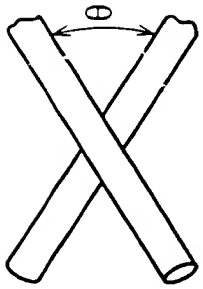


FIG._7B

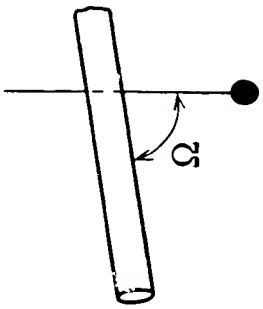


FIG._7D

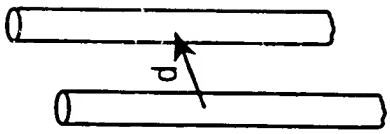


FIG._7A

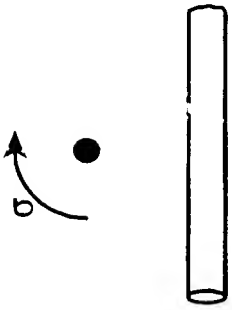


FIG._7C

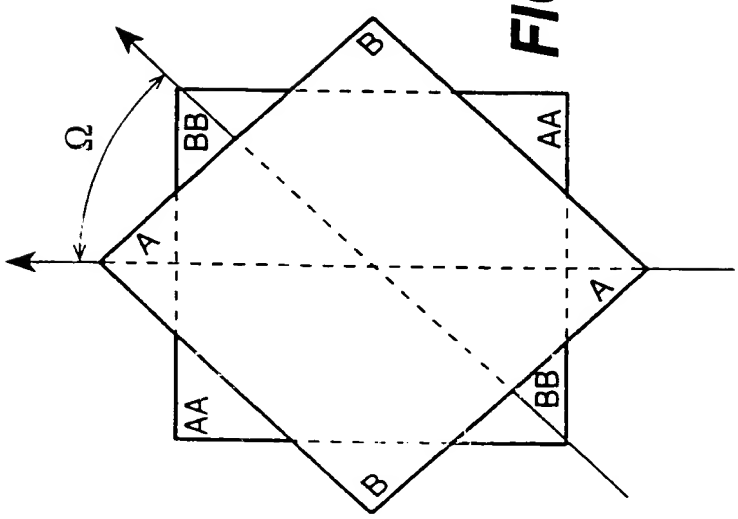


FIG._6C

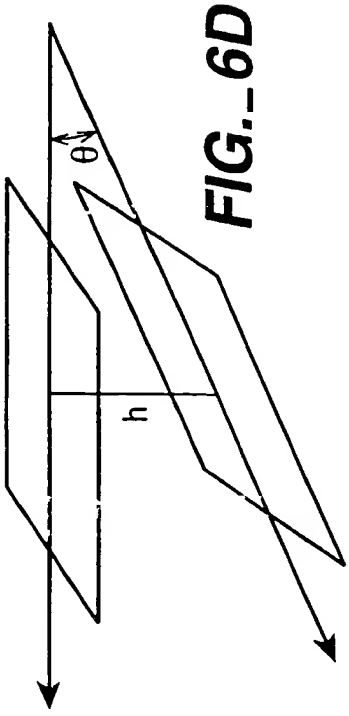


FIG._6D

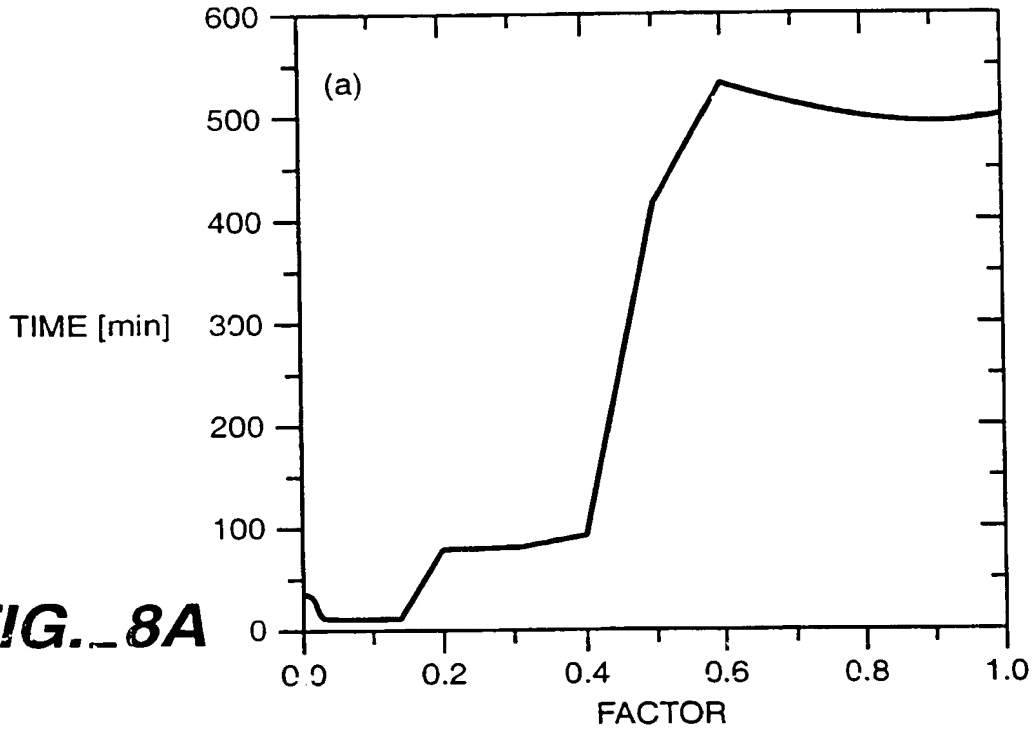

600
500
400
300
200
100
0
(a)
TIME [min]
0.0
0.2
0.4
0.6
0.8
1.0
FACTOR

FIG._8A

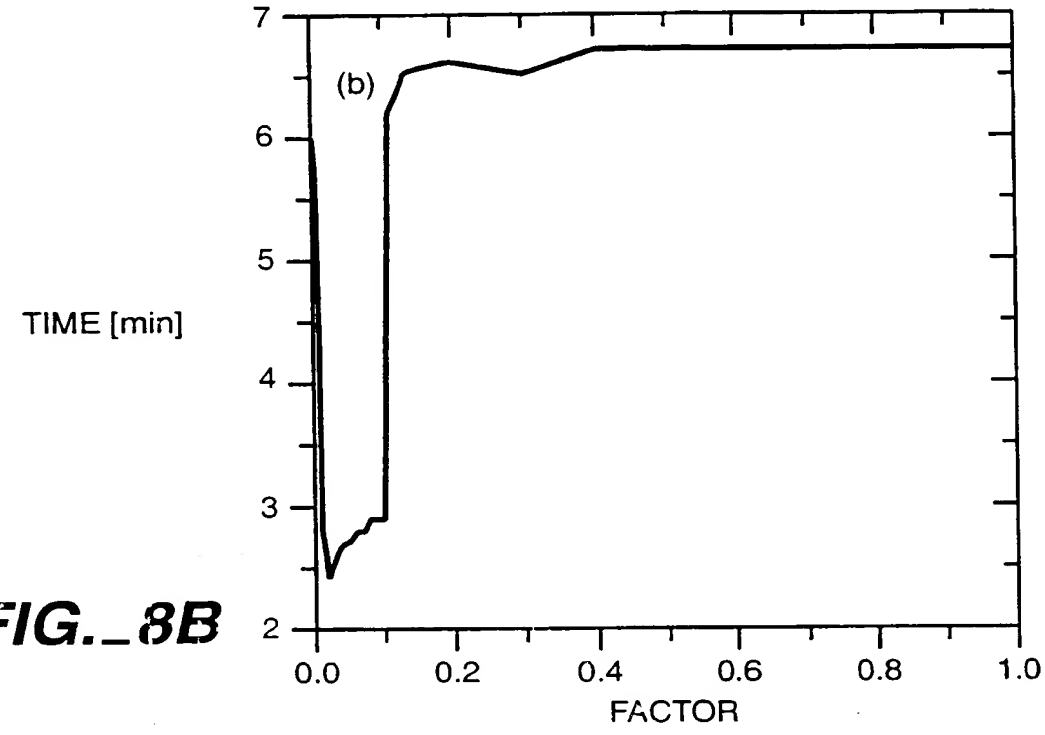

7
6
5
4
3
2
(b)
TIME [min]
0.0
0.2
0.4
0.6
0.8
1.0
FACTOR

FIG._8B

BENCHMARK TIMES					
	BENCHMARK CASES				
Total Times [min]	small C-B[a]	α-surface	β-surface	Mixture	Core-Boundary
DEE[b]	177.4	2.2	40.5	101.6	1154.0
B&B[c]	70.7	294.8	44.4	544.9	>30000[d]
B&T[e]	8.4	6.1	2.1	13.0	745.8
B&T Component Times [min]					
Preprocessing	0.1	0.1	0.1	0.4	0.6
Search	8.3	6.0	2.0	12.4	744.8
Approximation[1]				0.2	0.4
B&T Total Nodes	3829	1697	1546	845	34634

[a]Refers to the benchmark comprised of a small set of core and boundary positions.
[b]DEE was performed using the speed enhancements in Goldstein, R.F., Biophys. J., 66:1335-1340 (1994) and Gordon & Mayo, J. Comp. Chem., 19:1505-1514 (1998).
[c]The B&B algorithm uses the novel bounding expression and includes termination preprocessing.
[d]For the difficult Core-Boundary case, the incomplete B&B optimization was aborted after 30.000 minutes.
[e]Total B&T time is computed as the sum of the approximation, preprocessing, and search times.
[f]An approximate B&T algorithm was used to obtain initial bounds for the Mixture and difficult Core-Boundary cases. These calculations used only the top thirty rotamers at each position according to their bounding energy.

FIG._9

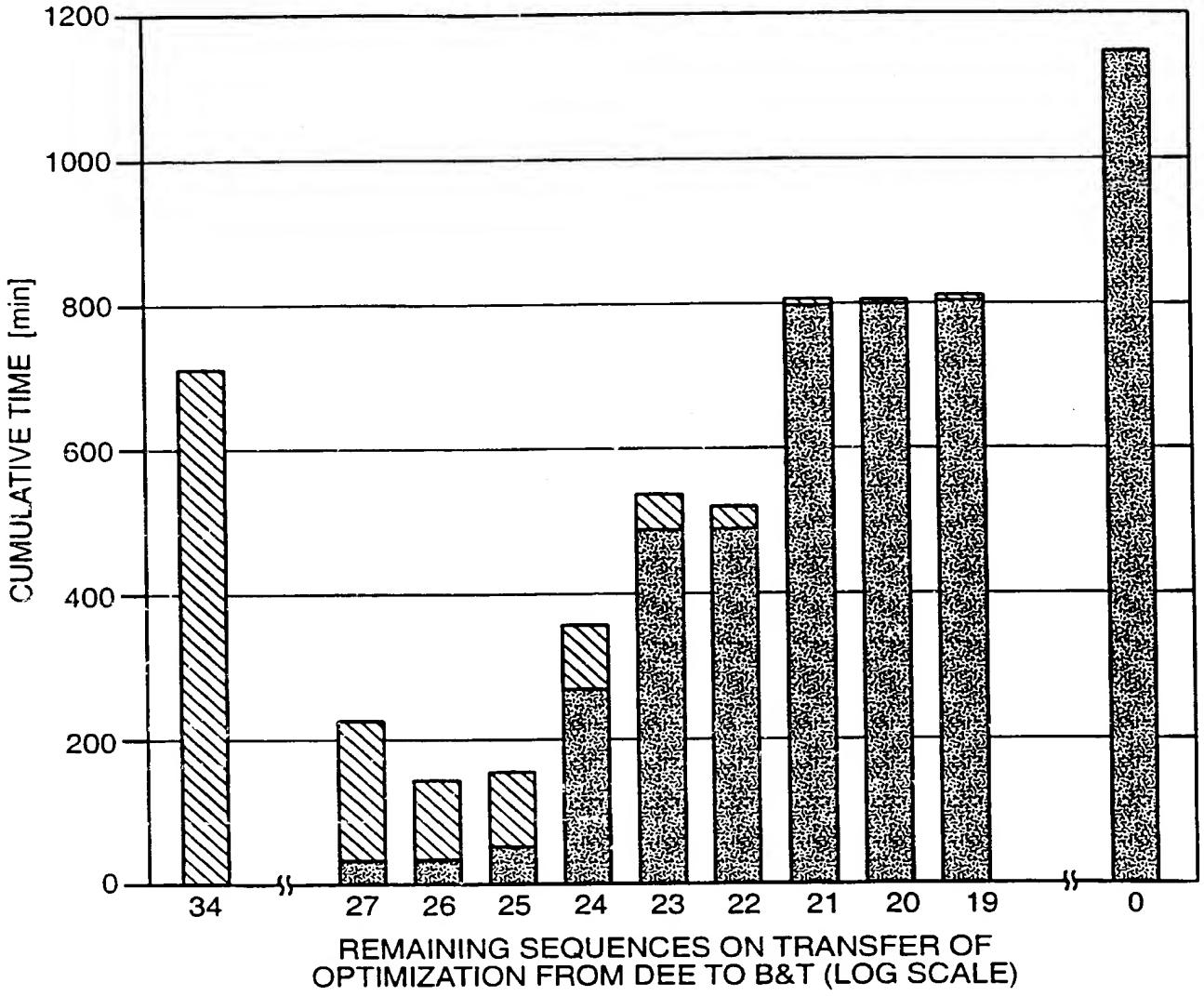

1200
1000
800
600
400
200
0
CUMULATIVE TIME [min]
34
27
26
25
24
23
22
21
20
19
0
REMAINING SEQUENCES ON TRANSFER OF
OPTIMIZATION FROM DEE TO B&T (LOG SCALE)

FIG._10

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
8 March 2001 (08.03.2001)

PCT

(10) International Publication Number
WO 01/16862 A3

(51) International Patent Classification[7]: G06F 19/00

(21) International Application Number: PCT/US00/40805

(22) International Filing Date:
1 September 2000 (01.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/151,818 1 September 1999 (01.09.1999) US

(71) Applicant: **CALIFORNIA INSTITUTE OF TECHNOLOGY** [US/US]; 1200 East California Boulevard, MC 201-85, Pasadena, CA 91125 (US).

(72) Inventors: **GORDON, David, B.**; 175 Beacon Street, Somerville, MA 02143 (US). **MAYO, Stephen, L.**; 530 S. Greenwood Avenue, Pasadena, CA 91107 (US).

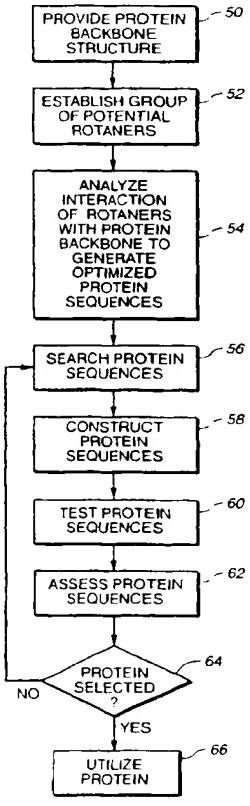
(74) Agents: **TRECARTIN, Richard, F.** et al.; Flehr Hohbach Test Albritton & Herbert LLP, 4 Embarcadero Center, Suite 3400, San Francisco, CA 94111-4187 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: METHODS AND COMPOSITIONS UTILIZING A BRANCH AND TERMINATE ALGORITHM FOR PROTEIN DESIGN



(57) Abstract: The present invention relates to apparatus and methods for quantitative protein design and optimization. In particular, the invention describes the use of the Branch and Terminate algorithm in protein design.

WO 01/16862 A3

Published:
- *with international search report*

(88) Date of publication of the international search report:
3 January 2002

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 00/40805

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F19/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,X	D GORDON AND S L MAYO: "Branch-and-Terminate: a combinatorial optimization algorithm for protein design" STRUCTURE WITH FOLDING & DESIGN, vol. 7, no. 9, 15 October 1999 (1999-10-15), pages 1089-1098, XP001028197 the whole document ---	1-27
Y	WO 98 47089 A (CALIFORNIA INST OF TECHN) 22 October 1998 (1998-10-22) abstract; claims 1-27 --- -/--	1-27

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 September 2001	26/09/2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Filloy García, E

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 00/40805

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No
Y	LATHROP R H AND SMITH T F: "A Branch-and-Bound Algorithm for Optimal Protein Threading with Pairwise (Contact Potential) Amino Acid Interactions" PROCEEDINGS OF THE TWENTY-SEVENTH HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, vol. V: Biotechnology Computing, 4 - 7 January 1994, pages 365-374, XP001027999 HI, USA abstract section 3 ---	1-27
A	US 5 680 331 A (SIANI MICHAEL A ET AL) 21 October 1997 (1997-10-21) abstract; claims 1-19 column 11, line 22 - line 29 ---	1-27
A	KLEPEIS JL ET AL: "Protein Folding and Peptide Docking: A Molecular Modeling and Global Optimization Approach" COMPUTERS & CHEMICAL ENGINEERING, vol. 22, 24 - 27 May 1998, pages S3-S10, XP001027996 UK abstract page S6, paragraph 3 - paragraph 7 -----	1-27

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 00/40805

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9847089 A	22-10-1998	AU 6965498 A	11-11-1998
		EP 0974111 A1	26-01-2000
		US 6188965 B1	13-02-2001
		WO 9847089 A1	22-10-1998
		US 6269312 B1	31-07-2001
US 5680331 A	21-10-1997	NONE	